64'er

APRIL 1984 ÖS 50.–/Sfr 6.– DM 6,–

4|84 DAS MAGAZIN FÜR COMPUTER-FANS

64, Farbmonitor und Floppylaufwerk in einem Gehäuse

Test 64 SX

Tragbares Kompaktsystem

Was nicht im Handbuch steht

Kurse zum Mitmachen:

6502-Assembler, Grafik für 64, Precompiler bauen, Codes richtig angewendet

Simons Basic macht das Programmieren leicht

Vergleichstest: Commodore-Drucker unter 1.000 Mark

Wettbewerbe:

Machen Sie mit beim Listing des Monats

Bargeld bis zu 2.000 Mark für Ihr Programm
1.000 Mark für das schönste Sprite

Was machen Sie mit Ihrem Computer? Jeder kann 500 Mark gewinnen

Unbekannte PEEKs und POKEs: Interessanten Adressen auf der Spur



Feldherr am Bildschirm: Strategiespiel „Caesar" ★ Erste Hilfe für gelöschte Programme ★ Grafik schnell gemacht: Sprites flott bewegt, Linien fix gezogen ★ ... und viele andere Programme mit ausführlicher Beschreibung für Commodore 64 und VC 20

Bitte Karte an der Perforation heraustrennen



64'er Mitmach-Karte

DAS MAGAZIN FÜR COMPUTERFANS IST DIE ZEITSCHRIFT ZUM MITMACHEN

☐ **Deshalb meine Meinung zu Heft /Seite /Artikel:** ______

☐ Ich wünsche mir für die nächsten Hefte folgende Themen: ______

☐ Ich stehe vor folgendem Problem: ______

☐ **Ich möchte mich an der redaktionellen Gestaltung von »64'er« beteiligen**
☐ Ich kann folgendes Programm zur Veröffentlichung anbieten
☐ Ich kann Ihnen über folgende Anwendung berichten

Bei Veröffentlichung meines Programmes/Berichtes erhalte ich ein angemessenes Honorar.

Bitte Karte an der Perforation heraustrennen

ANZEIGEN-AUFTRAG FÜR DIE 64'er FUNDGRUBE

★ ★

☐ **JA,** ich möchte die Gelegenheit nutzen und in der nächsten erreichbaren Ausgabe des Magazins »64'er« eine **private** Kleinanzeige für nur DM 5,- veröffentlichen. Der folgende Text (maximal 5 Zeilen mit je 32 Buchstaben) soll unter der Rubrik
☐ VC 20 ☐ Commodore 64 ☐ Software ☐ Zubehör erscheinen:

☐ Den Anzeigenpreis von DM 5,- habe ich auf das Postscheckkonto Nr. 14199-803 beim Postscheckamt München einbezahlt (Vermerk: Markt & Technik, »64'er«)
☐ DM 5,- in Briefmarken oder Bargeld liegen bei
☐ DM 5,- als Scheck liegen bei

Datum Unterschrift

★ ★

»64'er« ist die Zeitschrift zum Mitmachen. Bitte sagen Sie uns hier, ob und welchen Computer Sie haben, für welchen Sie sich interessieren, was Ihnen an »64'er« gefällt oder welche Themen Sie sich wünschen:

In dieser Ausgabe war besonders gut:

Für die nächsten Hefte wünsche ich mir folgendes Thema:

Ich besitze einen Computer: ☐ Ja ☐ Nein

Wenn ja: Welchen Computer: ____________

Wenn nein: Für welchen interessieren Sie sich, bzw. welchen wollen Sie kaufen?

Absender

Name/Vorname

Straße

PLZ/Ort

Postkarte

Antwort

Bitte frei- machen

64'er

Magazin für Computerfans

FUNDGRUBE

Markt & Technik
Verlagsgesellschaft mbH
Hans-Pinsel-Straße 2

8013 Haar bei München

»64'er« ist die Zeitschrift zum Mitmachen. Bitte beantworten Sie deshalb die folgenden Fragen: (Absenderangabe nicht vergessen):

In dieser Ausgabe war besonders gut: ____________

Ich besitze einen Computer: ☐ Ja ☐ Nein

Wenn ja: Welchen Computer: ____________

Wenn nein: Für welchen interessieren Sie sich, bzw. welchen wollen Sie kaufen?

Absender

Name/Vorname

Straße

PLZ/Ort

Postkarte

Bitte frei- machen

64'er
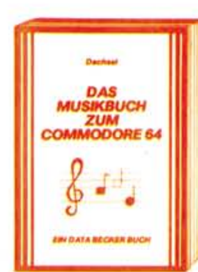
Magazin für Computerfans

REDAKTION

Markt & Technik
Verlag Aktiengesellschaft
Hans-Pinsel-Straße 2

8013 Haar bei München

NEUE SUPERHITS RUND UM COMMODORE



Der COMMODORE 64 ist ein Musikgenie, und mit diesem Buch lernen Sie alles über seine musikalischen Fähigkeiten. Der Inhalt reicht von einer Einführung in die Computermusik über die Erklärung der Hardware-Grundlagen und die Programmierung in BASIC bis hin zur fortgeschrittenen Musikprogrammierung. Zahlreiche Beispielprogramme und leicht verständliche Darstellung. Erschließen Sie sich die Welt des Sounds und der Computermusik mit dem MUSIKBUCH ZUM COMMODORE 64.
Ca. 200 Seiten, DM 39,–



Graphik ist eine der Hauptstärken des COMMODORE 64. Mit diesem neuen Buch lernen Sie, wie Sie die graphischen Fähigkeiten programmtechnisch optimal nutzen, von einfachen Figuren über Sprites, Zeichensatzprogrammierung, Hardcopy und IRQ-Handhabung bis hin zu Funktionendarstellung, Statistik, 3-D, CAD und den Geheimnissen der Actionsspiele. Zahlreiche Beispielprogramme ergänzen dieses Buch, das die faszinierende Computergraphik jedermann zugänglich macht.
Ca. 250 Seiten, DM 39,–



Umfassendes Nachschlagewerk zum COMMODORE 64 und seiner Programmierung. Allgemeines Computerlexikon mit Fachwissen von A–Z und Fachwörterbuch mit Übersetzungen wichtiger englischer Fachbegriffe – das DATA BECKER LEXIKON ZUM COMMODORE 64 stellt praktisch drei Bücher in einem dar. Es enthält eine unglaubliche Vielfalt an Informationen und dient so zugleich als kompetentes Nachschlagewerk und als unentbehrliches Arbeitsmittel. Ein Muß für jeden COMMODORE 64 Anwender!
Ca. 350 Seiten, DM 49,–



Nicht nur alles über Interfaces und Ausbaumöglichkeiten des COMMODORE 64 enthält dieses Buch, sondern auch über seine vielfältigen Einsatzmöglichkeiten von der Lichtorgel über Motorsteuerung, Spannungs- und Temperaturmessung bis zur programmierbaren Stromversorgung. Zehn komplette Schaltungen zum Selberbauen, vom Eprommer über den Digital-Voltmeter mit automatischer Messbereichswahl und den Logic Analyzer bis zur preiswerten Spracheingabe-Sprachausgabe. Jeweils mit Schaltplan, Layout und Softwarelisting.
Ca. 220 Seiten, DM 49,– ab April 84.



64 FÜR EINSTEIGER ist eine sehr leicht verständliche Einführung in Handhabung, Einsatz, Ausbaumöglichkeiten und Programmierung des COMMODORE 64, die keinerlei Vorkenntnisse voraussetzt. Schritt für Schritt führt das Buch Sie in die Programmiersprache BASIC ein, wobei Sie nach und nach eine komplette Adressenverwaltung erstellen, die Sie anschließend nutzen können. Das Buch ist sowohl als Einführung als auch als Orientierung vor dem 64er Kauf gut geeignet.
Ca. 200 Seiten, DM 29,–



Diese neue, umfangreiche Programmsammlung hat es in sich. Über 50 Spitzenprogramme für den COMMODORE 64 aus den unterschiedlichsten Bereichen, vom Superspiel über Graphik- und Soundprogramme sowie Utilities bis hin zu Anwendungsprogrammen. Der Hit sind zu jedem Programm aktuelle Programmiertips und Tricks der einzelnen Autoren zum Selbermachen. Also – nicht nur abtippen, sondern auch dabei lernen und wichtige Anregungen für die eigene Programmierung sammeln.
Ca. 250 Seiten, DM 49,–

... und natürlich die bewährten Bestseller aus bester Hand

Insgesamt über 200 000mal wurden die nachfolgenden Bücher in nur 12 Monaten verkauft. Machen auch Sie mehr als Ihrem COMMODORE mit diesen beliebten und bewährten Bestsellern aus bester Hand.



Endlich ein umfangreiches Trainingshandbuch, das Ihnen detailliert den Umgang mit SIMON's BASIC erklärt. Ausführliche Darstellung aller Befehle und ihrer Anwendung. Zahlreiche Beispielprogramme und Programmiertricks. Dieses Buch sollte jeder SIMON's BASIC Anwender haben!
ca. 300 S., DM 49,–

Eine leicht verständliche Einführung in das Programmieren des C-64 in Maschinensprache und Assembler. Komplett mit vielen Beispielen sowie einem Assembler, Disassembler und einem Einzelschritt-Simulator. Und natürlich zugeschnitten auf Ihren Computer, den COMMODORE 64.
ca. 200 S., DM 39,–

64 INTERN erklärt detailliert Technik und Betriebssystem des C-64 und die Programmierung von Sound und Graphik. Ausführlich dokumentiertes ROM-Listing, zahlreiche lauffertige Beispielprogramme und 2 Original-Schaltpläne zum Ausklappen. Dieses Buch sollte jeder 64-Anwender und Interessent haben.
ca. 320 S.; DM 69,–

64 TIPS & TRICKS ist eine echte Fundgrube für jeden COMMODORE 64 Anwender. Umfangreiche Sammlung von POKE's und anderen nützlichen Routinen, BASIC-Erweiterungen, Graphik und Farbe für Fortgeschrittene, CP/M, Multitasking, mehr über Anschluß- und Erweiterungsmöglichkeiten und zahlreiche lauffertige Programme.
ca. 290 S.; DM 49,–

64 FÜR PROFIS zeigt, wie man erfolgreich Anwendungsprobleme in BASIC löst und verrät Erfolgsgeheimnisse der Programmierprofis. 5 komplett beschriebene, lauffertige Anwendungsprogramme (z. B. Adreßverwaltung) illustrieren den Inhalt der einzelnen Kapitel beispielhaft. Mit diesem Buch lernen Sie gute und erfolgreiche BASIC-Programmierung.
ca. 320 S., DM 49,–

DAS GROSSE FLOPPY-BUCH erklärt detailliert die Arbeit mit der Floppy VC-1541, von der sequentiellen Datenspeicherung bis zum Direktzugriff, für Anfänger, Fortgeschrittene und Profis. Ausführlich dokumentiertes DOS-Listing, zahlreiche lauffertige Beispiel- und Hilfsprogramme, z. B. Disk Editor und Haushaltsbuchführung.
ca. 320 S.; DM 49,–

VC-20 INTERN ist für jeden interessant, der sich näher mit Technik und Maschinenprogrammierung des VC-20 auseinandersetzen möchte. Detaillierte technische Beschreibung des VC-20, ausführliches ROM-Listing, Einführung in die Maschinenprogrammierung und 3 Original-Schaltpläne.
ca. 230 S.; DM 49,–

VC-20 TIPS & TRICKS ist eine echte Fundgrube für jeden VC-20 Anwender. Sound und Graphik Programmierung, Speicherbelegung und Speichererweiterung, BASIC-Erweiterungen, POKE's und andere nützliche Routinen, zahlreiche lauffertige Beispiel- und Anwendungsprogramme und vieles andere mehr.
ca. 230 S.; DM 49,–

Darauf haben Sie gewartet!



Die neue DATA WELT ist eine Computerzeitschrift speziell für COMMODORE-Anwender. Brandaktuell (detaillierte Informationen über die neuen COMMODORE Computer 264 und 364) und randvoll mit Berichten, Trends und interessanten Programmiertips. 80 Seiten stark im Magazinformat. Gleichzeitig als Nachfolger des VC-Infos umfassende Übersicht über aktuelle Produkte, Bücher und Programme rund um COMMODORE 64 und VC-20. Die Frühjahrsausgabe der neuen DATA WELT erhalten Sie ab Anfang März überall dort, wo es DATA BECKER BÜCHER und -PROGRAMME gibt. Am besten gleich holen oder direkt bei DATA BECKER gegen DM 4,– in Briefmarken anfordern.

IHR GROSSER PARTNER FÜR KLEINE COMPUTER

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010 · im Hause AUTO BECKER

DATA BECKER BÜCHER und PROGRAMME erhalten Sie im Computer-Fachhandel, in den Computerabteilungen der Kauf- und Warenhäuser und im Buchhandel. Auslieferung für Österreich Fachbuch-Center ERB, Schweiz THALI AG und Benelux COMPUTERCOLLECTIEF.

CP

BESTELL-COUPON

Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1

Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
☐ DATA WELT 1/84 (DM 4,– in Briefmarken liegen bei)

Name und Adresse bitte deutlich schreiben

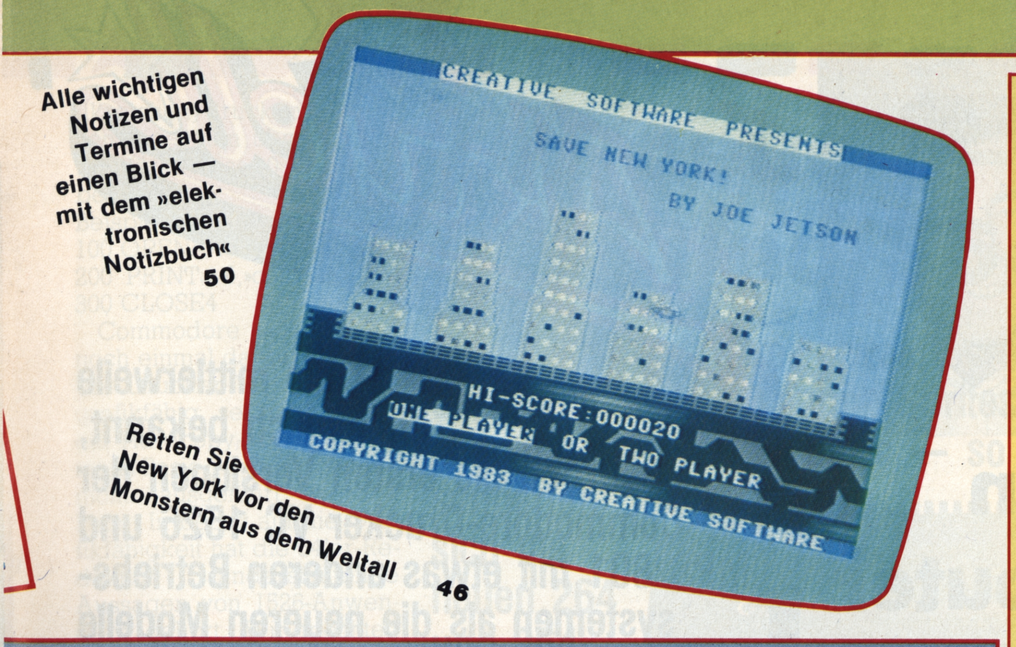
INHALT



Relativ schnell stößt man an die Grenzen seines Computers. Da hilft nur noch der Kauf von Erweiterungen. Wir sagen Ihnen, wo es welche gibt 34

Der Computer hilft beim Auswerten von Wettkämpfen — mit komplettem Listing 56

Alle wichtigen Notizen und Termine auf einen Blick — mit dem »elektronischen Notizbuch« 50



Retten Sie New York vor den Monstern aus dem Weltall 46

Jetzt gibt es auch von Commodore einen Tragbaren — den 64 SX. Er ist zum Commodore 64 voll kompatibel und hat einen eingebauten Farbmonitor und ein Floppy-Laufwerk 27

COMMODORE DRUCKER IM TEST
ZEITVERGLEICH

Ohne Drucker kann man nicht vernünftig arbeiten. Doch welcher eignet sich wofür am besten? 20

4/84



Speziell für Ihren Computer

Zu Jahresbeginn standen in der Bundesrepublik etwa 125000 Commodore 64 und gut 140000 VC 20. Das ist nicht nur ein Zeichen dafür, daß die Heimcomputer im vergangenen Jahr den Durchbruch in Deutschland geschafft haben. Es war zugleich auch an der Zeit, daß es speziell für die Benutzer dieser Tastatur- oder Volkscomputer, die den größten Marktanteil haben, eine eigene Zeitschrift gibt.

64'er wird sich ausschließlich mit VC 20 und Commodore 64 befassen — und mit den weiteren Mitgliedern dieser Computerfamilie, die ja schon angekündigt sind. Spezialisierung zum Vorteil des Lesers: Er findet hier im Detail alle Informationen über »seine« Systemfamilie — und ist andererseits sicher, daß er die Programmiertips und Listings aus jedem Heft direkt verwenden oder zumindest für sich auswerten kann (eine Verwendbarkeit jeder Information für alle Computer ist wegen der modellbedingten Unterschiede nicht zu erreichen). 64'er soll aber auch zum Forum für alle Benutzer der kleinen Commodore-Systeme werden: Wir hoffen, daß sich möglichst viele Leser in der einen oder anderen Form an der Gestaltung dieser Zeitschrift beteiligen. Das kann in Form von Anfragen, von Kritik oder von Antworten auf Fragen anderer Leser geschehen, durch Beteiligung an unseren Wettbewerben oder dadurch, daß Sie uns — es gibt natürlich Honorar dafür — Artikel oder Listings zur Veröffentlichung anbieten. Jeden Monat geht es da um eine ganze Menge Geld: um 2000 Mark für das Listing des Monats (Seite 127), um 500 Mark für die Anwendung des Monats (Seite 127), und um viele Hundertmarkscheine, die den Autoren interessanter Programme winken (Seite 126). Außerdem haben wir diesmal einen Tausender für denjenigen ausgesetzt, der das schönste Sprite entwickelt hat (Seite 126); weitere derartige Wettbewerbe werden folgen. Wer bei einem internationalen Spiele-Programmierwettbewerb, bei dem es allein an Preisen 17500 Dollar zu gewinnen gibt, mitmachen will, findet Informationen über die Teilnahmebedingungen auf der Seite 64.

Sogar für diejenigen, die »nur« ein Problem haben, lohnt sich das Mitmachen: Im Leserforum veröffentlichen wir regelmäßig Fragen, die unsere Leser stellen — entweder gleich mit einer Antwort oder in der Hoffnung, daß aus dem Leserkreis eine Antwort eingeht, die dann in einem späteren Heft publiziert wird.

Wir setzen darauf, daß 64'er »Ihre« Zeitschrift wird — und daß Sie dabei mitwirken. Damit Sie es leichter haben, finden Sie in jedem Heft eine »Mitmach-Karte«.

Michael Pauly, Chefredakteur

Aktuell

Wie mittlerweile hinlänglich bekannt, wurden die ersten Versionen der Commodore-Drucker VC 1526 und MPS 801 mit etwas anderen Betriebssystemen als die neueren Modelle geliefert. Was hat sich nun geändert, und inwieweit sind Programme, die für die älteren Systeme erstellt wurden, zu modifizieren?

1526 und MPS 801 - Was ist neu?

Im neuen Betriebssystem des Druckers VC 1526 sind zum einen die Fehler des alten behoben und zum anderen ist eine zweite Betriebsart (die des 1525) implementiert worden. Das mitgelieferte Handbuch bezieht sich auf die zweite Betriebsart des Druckers, die aber nur hardwaremäßig eingestellt werden kann. Dies geschieht etwas umständlich, indem man PIN 16 von U4D auf Masse legt. In folgenden Punkten unterscheidet sich die standardmäßige Betriebsart (die beim Einschalten des Druckers vorliegt) von den Steuerbefehlen, die im Druckerhandbuch aufgeführt sind.

☐ Die Breitschrift wird mit CHR$(14) ein- und mit CHR$(15) ausgeschaltet.

☐ Es ist ein neuer Steuerbefehl hinzugekommen. Mit CHR$(16) wird der Tabulator auf eine bestimmte Spalte gestellt.

Beispiel:

```
100 OPEN4,4
200 PRINT# 4,CHR$(16)«08«
300 PRINT# 4, »64'er — Das Magazin für Computer-Fans«
400 CLOSE4
```

Durch die Zeile 200 wird der Druckstart auf die achte Spalte festgelegt.

☐ Die Funkion der Sekundäradresse 7 als Schalter wurde aufgegeben. Stattdessen werden die Daten über die Sekundäradresse 7 ausgegeben.

Beispiel:

```
100 OPEN 7,4,7
200 PRINT# 7, »Das 64'er bietet alle Informationen für den Commodore-Anwender«.
300 CLOSE7
```

☐ Durch diese letzte Änderung entfällt die Sekundäradresse 8 vollständig. Will man nun im Grafikmodus (Großbuchstaben und Gra-

fikzeichen) drucken, so müssen die Daten über die Sekundäradresse 0 ausgegeben werden.
Beispiel:
100 OPEN4,4
200 PRINT# 4,»commodore«
300 CLOSE4

Commodore selbst weist noch einmal darauf hin, daß der Drucker VC 1526 nicht grafikfähig sei und daher ein Hardcopy-Befehl mit Simons Basic (High Resolution) nicht ausgeführt werden kann. In bezug auf die Grafikfähigkeit hat die 64'er-Redaktion anderslautende Aussagen von 1526-Anwendern gehört. Wir möchten unsere Leser aufrufen, sofern sie eine funktionierende Hardcopy-Routine entwickelt haben, diese mit einer ausreichenden Beschreibung an uns zu schicken. Die besten und schnellsten werden prämiert und veröffentlicht.



Was hat sich nun bei der Änderung im Betriebssystem des jüngsten Mitglieds der Commodore Druckerfamilie, dem MPS 801 (MPS steht übrigens für Matrix Printer System) ergeben?

- Nur die Sekundäradressen 0 und 7 haben ihre Bedeutung beibehalten, alle anderen Sekundäradressen sind entfallen.
- CHR$(141) hat jetzt die Funktion »Sperrschrift ein« übernommen.
- Der Character-Code für »Bit-Muster aus« und für »Sperrschrift aus« wurde durch CHR$(15) für »Standard-Zeichensatz ein« ersetzt.
- CHR$(141) für »CR ohne Zeilenvorschub« entfiel vollständig.

Durch diese Änderungen sind natürlich die Programmbeispiele in den Druckerhandbüchern nicht mehr up-to-date. Sollten Ihnen noch weitere Änderungen bekannt geworden sein, so lassen Sie es uns wissen. (aa)

Die neuen — 264 und 364

Eingebaute Software, etwas teurer als der Commodore 64, ohne Sprites aber mit »besseren« Basic — so präsentierten sich die neuen 264 und 364 auf der Consumer Electronics Show (CES) in Las Vegas, der größten Unterhaltungselektronik-Messe in den USA.



Eines gilt als sicher: Ablösen sollen die 264/364 den Commodore 64 nicht, sondern »nach oben« ergänzen. Jim Butterfield, in den USA bekannt als die (unabhängige) Commodore-Autorität stufte den 264/364 innerhalb der Reihe der kleinen Commodore-Systeme wie folgt ein: »Der VC 20 ist ein Computer mit einem niedrigen Preis speziell für Leute, die ihre ersten Schritte mit einem Computer tun wollen. Der 64 ist der »Fun-Computer«. Er unterstützt den creativen Programmierer, hat tolle Sound- und Grafikmerkmale und ist aufgrund seiner internen Stuktur für alle Arten von Erweiterungen geeignet. Er läßt dem Anwender einen großen Spielraum, was er mit dem 64 alles tun kann. Und das ist die Meinung des »Guru« zum 264/364: »Der 264/364 geht mehr in Richtung Business. Er hat Merkmale, die sowohl dem Programmierer als auch dem reinen Benutzer helfen, mit dem Computer schnell und einfach zu arbeiten. Das eingebaute Basic ermöglicht das Programmieren von kaufmännisch-orientierter Software und von Grafik. Die neue Tastatur mit einer Hilfe-Taste und mehreren Funktionstasten tut ein übriges. Und«, fügte Butterfield hinzu, »die eingebauten professionellen Programme erlauben es, sofort nach dem Einschalten mit dem 264/364 zu arbeiten«.

Typischerweise sollen sie zu Hause für ernsthafte Dinge eingesetzt werden oder den »kleinen Geschäftsmann bei seiner täglichen Arbeit unterstützten«.

Wesentliche Unterschiede zwischen Commodore 64 und den »Neuen«: Die »Neuen« werden mit »eingebauter« Software und einem erweiterten Basic geliefert, verfügen jedoch nicht über Sprites, die beim Commodore 64 so beliebten, selbstdefinierbaren Grafikelemente. Die Sache mit der »eingebauten« Software funktioniert so: Beim Kauf eines Commodore 264/364 kann man — so der momentane Stand — zwischen vier Programmen (3 plus 1, Superscript, Magic Desk und Logo), wählen. Der Computer wird dann mit dem bereits

Apple Software für Commodore 64?

Der Hardware-Zusatz »AP Modular Pak« von Mimic Systems, Kanada, soll den Zugriff auf das riesige Angebot von Apple II-kompatibler Peripherie und Software erlauben. Mit dem AP Modular Pak soll jedes für den Apple II entworfene Programm auf dem Commodore 64 laufen. Jede Apple II-kompatible Hardware funktioniert genauso, als ob sie an den Apple II angeschlossen wäre. Das bedeutet, daß die verschiedenen für den Apple II verfügbaren Prozessoren jetzt mit dem Commodore 64 benutzt werden können. Das AP Modular Pak besitzt drei Komponenten: den AP-Bus, der acht Standard-Apple II-Peripherie-Steckplätze und vier Commodore 64-Erweiterungs-Steckplätze enthält, dann die AP »CPU«-Karte, die in einem eigenen Steckplatz auf dem AP-Bus steckt und alle Umwandlungen vom Apple II zum C64 bewältigt, und drittens die AP-DOS-Karte, eine Peripherie-Karte für die Floppy VC 1541 von Commodore, die die VC 1541 in ein preiswertes, Apple II-kompatibles Laufwerk umwandelt. Preis: unter 500 Dollar.

»eingebauten« Programm ausgeliefert. Die anderen Programme können zwar auch eingesetzt werden, allerdings nur in Form der vom Commodore 64 bekannten Software-Module (Cartridges).

Doch nun zu den technischen Merkmalen der neuen Modelle 264/364:

Von den 64 KByte RAM stehen dem Benutzer 60 KByte für Basic-Programme zur Verfügung. In ROMs sind Betriebssystem und Basic-Interpreter sowie die eben erwähnte, »eingebaute« Software untergebracht. Der neue Mikroprozessor 7501 ist voll 6502-kompatibel und verfügt darüber hinaus noch über einige zusätzliche Fähigkeiten. Der Commodore 264/364 kann insgesamt 128 Farben (16 Farben; 8 verschiedene Stufen) darstellen. In 25 Zeilen zu je 40 Zeichen können Groß- und Kleinbuchstaben, Zahlen und Symbole normal, invers oder blinkend untergebracht werden. Der Zeichensatz-ROM enthält auch sämtliche bekannte Commodore-Grafikzeichen. Der Bildschirm kann in drei verschiedene Darstellungsarten geschaltet werden: Text, hochauflösende Grafik und Text/hochauflösende Grafik gemeinsam. Die Auflösung beträgt 320 x 200 Bildpunkte. Zwei Tongeneratoren oder ein Ton- und ein Geräuschgenerator sind in acht Stufen programmierbar.

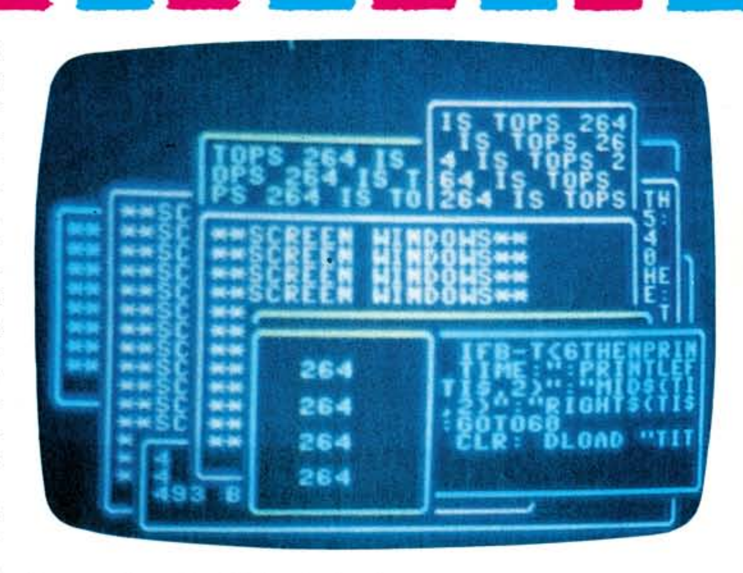
Die schreibmaschinenähnliche Tastatur besteht aus insgesamt 67 Tasten, darunter vier pfeilförmigen Cursortasten, vier programmierten beziehungsweise reprogrammierbaren Funktionstasten. An Schnittstellen stehen zur Verfügung: User Port und serielle Schnittstelle (identisch mit Commodore 64), Modul-(Cartridge-) Port zwei Joystickanschlüsse, Anschlüsse für Datasette (alle nicht identisch mit Commodore 64) und Fernsehgerät sowie für Farbmonitor, Tonein- und -ausgang und Stromversorgung.

Das eingebaute Basic 3.5 verfügt über 75 Befehle; es enthält das vollständige 64er-Basic und darüber hinaus noch zusätzliche Befehle. Weitere Software-Merkmale: Maschinensprache-Monitor mit über 12 Befehlen, Grafik- und Sound-Kommandos sowie eingebautes Windowing (der Bildschirm kann in einzelne Bereiche — Fenster, englisch: window — eingeteilt werden, die unabhängig voneinander beschrieben oder gelöscht werden können).

Die Commodore 264/364 arbeiten mit folgenden, bereits vom 64 her bekannten Peripheriegeräten: Floppy-Disk-Laufwerk 1541, Farbmonitor und Matrixdrucker 1526.

Nur mit den neuen Commodore-Computern sind die modifizierte Datasette (Kassettenlaufwerk) 1531 und das fünf- bis sechsmal schnellere Diskettenlaufwerk SFS 481 einsetzbar. Das neue Diskettenlaufwerk SFS 481 soll aber auch für den Commodore 64 angeboten werden. Ansonsten gibt es für die Commodore 264/364 noch den Farb-Matrixdrucker MCS 801 und den Typenraddrucker DPS 1101 (18 Zeichen pro Sekunde; die Typenräder sind Triumph-Adler-kompatibel).

Der 364 unterscheidet sich vom 264 nicht nur durch ein größeres Gehäuse (42 cm x 6,5 cm 24 cm gegenüber 33,5 cm x 6 cm x 19,5 cm), sondern auch durch ein eingebautes Sprachmodul (250 Worte Standard, zusätzliche Worte können von Modulen oder Diskette zugeladen werden), größere Tastatur (86 Tasten anstatt 67 — die zusätzlichen 19 Tasten stellen einen Ziffernblock dar), einen größeren ROM-Bereich (48 KByte anstatt 32 KByte) und zusätzliche Befehle für das Sprachmodul. Auch der ROM-Bereich für »eingebaute« Software ist größer: 48 KByte statt 32 KByte.



Im neuen Commodore 264 bereits eingebaut: Windows (Fenster)



Ob der Commodore 116 jeweils auf den Markt kommen wird, ist zweifelhaft

Die 264/364 sollen ab April in den USA erhältlich sein. Commodore selbst gibt keinen genauen Preis an, aber 300 bis 400 Dollar dürften realistisch sein. Im Unterschied zum 64 sei bei den »Neuen« das Basic besser, Hauptverkaufsargument ist die built-in-Software (»eingebautes« Programm). Da die Programm-Module nicht identisch sind, können die des Commodore 64 im 264/364 nicht verwendet werden.

Zur Frage der Kompatibilität: 64-Programme ohne Sprites, laufen auf den »Neuen«; die restliche Software soll zu 80 bis 90 Prozent ohne oder mit »nur geringem« Umstellungsaufwand auch für den Commodore 264/364 verfügbar sein. 64er-Programme, die auf Kassette vorliegen, können nicht in die »Neuen« geladenen werden, bei Programmen von Diskette soll es jedoch möglich sein — das bedeutet aber noch lange nicht, daß die Programme dann auch auf dem 264/364 laufen.

Sicher scheint nur eines zu sein: Der 264/364 soll den Commodore 64 — ein Sprachmodul gibt es jetzt auch für den 64 für knapp 60 Dollar — nicht ablösen, sondern nur ergänzen. Das Sprachmodul für den 64 enthält 235 fest eingebaute

Höhenflüge

Die Ewings und Carringtons würden angesichts der Umsatzsteigerungen bei Data Becker vor Neid erblassen. Wie kaum ein anderes Unternehmen haben es diese beiden verstanden, das Wirtschaftswunder der fünfziger Jahre auch 1983 noch einmal wahr werden zu lassen.

Kräftig mitgeschwommen (aber auch mitgerudert) ist Data Becker dabei auf der Erfolgswelle des Commodore 64. Eine Umsatzsteigerung von nahezu 300 Prozent (von 8 Millionen Mark 1982 auf 23 Millionen Mark 1983) spricht eine deutliche Sprache.

Der Heimcomputermarkt erreichte 1983 nach Schätzungen

Worte, gesprochen von einer »angenehmen Frauenstimme«, soweit eine Commodore-Mitteilung. Die einzelnen Worte können direkt von Basic oder Assembler angesprochen werden. Auch die Sprechgeschwindigkeit kann gewählt werden — langsam, normal oder schnell. Ein seperater Ausgang sorgt für eine Verbindung zur Hifi-Anlage. Ein zusätzlicher Wortschatz und verschiedene Stimmen auf Diskette oder Kassette wurden bereits angekündigt. Dieses 60-Dollar-Modul paßt auch für den tragbaren SX 64.

Ein weiterer Neuer, der wohl zumindest in den USA — nie seinen Weg über den Ladentisch finden wird: Der Commodore 116. Bei Commodore-USA selbst gab's zu diesem Computer weder Informationen noch eine Stellungnahme.

Auch der VC 20 soll weiterhin produziert werden, »solange er gekauft wird« (Preisvorstellungen: 100 Dollar weniger als der 64). Das ist auch nicht verwunderlich, wenn man bedenkt, daß man mit allen Modellen 1983 Rekordumsätze machte, und daß mittlerweile über zwei Millionen Commodore 64 verkauft wurden (die Jahresvorgabe von 1 Million wurde in einem halben Jahr abgesetzt) und Commodore 1983 1 Milliarde Dollar Umsatz machte.

Commodore hat vor kurzem mit Compuserve eine Vereinbarung über die weltweite Vermarktung des Vidtex Terminal Emulators geschlossen. Dieses Programm erlaubt via Modem (Vicmodem für zirka 60 Dollar und Automodem für zirka 100 Dollar — insgesamt wurden davon in den USA 1983 100000 Stück verkauft; in Deutschland sind sie noch nicht erlaubt, da eine FTZ-Zulassung noch nicht vorliegt), sich Programme aus der großen Compuserve-Software-Bibliothek zu holen und auf Diskette abzuspeichern. Außerdem ist es mit Vidtex möglich, aus dem »Commodore-Informations-Netz« technische Informationen, kostenlose Software und eine kostenlose »elektronische« Zeitschrift zu beziehen sowie mit anderen Benutzern in Kontakt zu treten.

Bookware baut Commodore kräftig aus: das Programm umfaßt Computerbücher, Bücher mit beiliegender Software und Computerzeitschriften. Wie gut »Bookware« bei Commodore-Benutzern ankommt, zeigen VC 20 Programmer's, Reference Guide und Commodore 64 Programmer's Reference Guide von denen 1983 jeweils 600.000 Stück verkauft wurden. Auch für die Commodore 264/364 soll es solche »Programmierführer« geben. (sc)

2 Joystick
Ein/Aus Schalter
Power
Cartridge
Abstimmung
Fernseher
Video
serielle Schnittstelle
Datasette
Userport

Power (wie bei Commodore
serielle Schn. 64) (wie bei Commodore 64)
Datasette (entspricht nicht Commodore 64)
User Port (wie bei Commodore 64)
Cartridge (entspricht nicht Commodore 64)
Joystick (nicht kompatibel zu Commodore 64)
Video (entspricht Commodore 64)

Die Anschlüüse des C 64 (oben) und die neuen des 264 (unten)

der Düsseldorfer Data Becker GmbH ein Volumen von rund 350000 Stück. Vor allem im letzten Quartal des Jahres habe sich ein Superboom entwickelt. Dieser werde aber vor allen Dingen durch den im Heimcomputermarkt fast konkurrenzlosen Commodore 64 getragen. Dabei habe sich ein eindeutiger Trend vom Wegwerf-Computer hin zum universell einsetzbaren, leistungsfähigeren Gerät gezeigt. Diese universelle Einsetzbarkeit und eine sehr vielseitige Software werden nach Aussagen von Firmenchef Dr. Achim Becker dazu beitragen, daß der Heimcomputer keine kurzfristige Modeerscheinung wird, sondern ein Produkt mit einer großen und langen Zukunft. Parallel zu den Heimcomputern boomt natürlich auch der Markt für entsprechende Peripheriegeräte, Zubehör und vor allem der Markt für Computerbücher und Software.

Dementsprechend hat sich das Verlagsgeschäft mit eigenen Computerfachbüchern zu einem wesentlichen Umsatzträger von Data Becker entwickelt. Ende 1983 belief sich die Zahl der verkauften Bücher — bei sieben Titeln — auf 150000. Für 1984 ist mit einem auf etwa 30 bis 40 Titeln anwachsenden Sortiment ein Mindestumsatz von 500000 Büchern geplant. Am Gesamtumsatz habe das Buchgeschäft bereits einen Anteil von 22 Prozent erreicht.

Im Bereich der Software gibt Data Becker mit neuen auf den Commodore 64 zugeschnittenen Paketen einen Verkauf von monatlich 10000 Programmen an, das entspricht einem Anteil von sieben Prozent am Gesamtumsatz. Trotz dieser Verkaufszahlen geht Data Becker energisch gegen den Softwareklau und die Raubkopierer vor. Über 200 Verfahren sind bereits eingeleitet worden. In den bisher zur Entscheidung gekommenen Rechtsstreitigkeiten hat Data Becker immer gewonnen. (aa)

Das Neueste aus USA

Dreißig Programme sollen verfügbar sein, wenn der neue Commodore 264/364 im April dieses Jahres auf den US-Markt kommt. Aber auch für den Commodore 64 gibt es neue Software und Erweiterungen.





Gibt es für Commodore 64 und die neuen 264/364: Magic-Desk — Bürolandschaft auf dem Bildschirm

Als Modul oder Diskette soll es ab Frühjahr für 64/264/364 geben: Magic Desk (Textverarbeitungs-, Tabellenkalkulations-, Dateiverwaltungsprogramm und Rechenfunktionen für zu Hause), 3-Plus-1 (Textverarbeitung, Dateiverwaltung, Tabellenkalkulation und Grafik mit Window-Fähigkeiten, das heißt der Textverarbeitungs- und der Tabellenkalkulationsteil können gleichzeitig am Bildschirm dargestellt und bearbeitet werden) und Superscript (professionelles Textverarbeitungsprogramm mit Serienbrief-Funktion). Diese drei Programme sind wahlweise auch »eingebaut« in einen Commodore 264/364 erhältlich. Das vierte, ebenfalls »einbaubare« Programm ist Logo, das auf Diskette 80 Dollar kosten soll. Ein Paket, bestehend aus Commodore 64, Farbmonitor, Diskettenlaufwerk 1541 und Logo soll für unter 800 Dollar angeboten werden.

Weitere Programme, die für Commodore 64 und 264/364 geplant sind: Micro Illustrator (Mal- und Zeichenprogramm, das es bislang unter anderem für Atari, Chalkboard und Koala gibt und bereits 100000 mal verkauft wurde), A Bee C's, Counting Bee, Gorf, Wizard of War-(Lern-)Spiele, die von dem Sprachmodul »Magic Voice« gebrauch machen, Easycalc 64 und Easycalc 264 (Tabellenkalkulationsprogramm mit Farb- und Grafik-Möglichkeiten) und B/Graph (Grafik und Statistik).

Darüber hinaus soll es in diesem Jahr noch eine Reihe von Programmen vorerst nur für die Commodore 64-Besitzer geben: »Commodore Kids«, eine Serie von Lernprogrammen für zu Hause, ferner die hauptsächlich mathematisch orientierte Lernspiel-Serie Milliken Edufun sowie die »Kinder Koncepts Series«, eine Sammlung von 40 Programmen auf fünf Disketten für Kinder im Alter von 4 bis 6 Jahren. Die letztgenannte Serie enthält hauptsächlich Programme aus dem Bereich Lesen- und Rechnenlernen. Des weiteren sind eine Reihe von Spielen, die »die Fähigkeiten der Computer ausnutzen und von den Videospielen herkömmlicher Art weit entfernt sind«, wie man bei Commodore beteuert, geplant. Dazu gehören International Soccer (dreidimensionales Fußballspiel; zirka 35 Dollar), Viduzzles (Video-Puzzles für Kinder), Jack Attack (Strategiespiel) und Solar Fox (Abenteuerspiel). Den Computer für die Hausarbeit einsetzen — das soll mit dem Micro Cookbook möglich sein. Micro Cookbook erlaubt das Planen und Zusammenstellen von Mahlzeiten — eine Art elektronisches Kochbuch mit Rezepten, Zutaten, Fachbegriffverzeichnis und Kalorientabelle. Preis: unter 40 Dollar.

Für den Manager 64, eine Art Datenbankprogramm, gibt es jetzt eine Reihe von vorgefertigten Applikationen aus allen möglichen Bereichen des täglichen Lebens. Verfügbar sind »Home Manager«, »Spots Manager« und Business Manager; weitere sind bereits geplant.

Auf Erweiterungen für den Commodore 64 und den VC 20 hat sich beispielsweise Cardeo spezialisiert. Das Angebot reicht von Expansionboards über Druckerinterfaces bis zu Lichtgriffel. Besonders interessant für Commodore-64-Freunde ist sicher das Cardboard 5, eine Erweiterung des Cartridge(Modul-)Steckplatzes um fünf, mit Schaltern anwählbare, Steckplätze (zirka 80 Dollar). Von MSD gibt es für die gleichen Rechner neben Erweiterungskarten und diversen Schnittstellen auch anschlußfertige 5¼-Zoll-Diskettenlaufwerke (170 KByte). Preis: 399 Dollar für ein Laufwerk, 695 Dollar für ein Doppellaufwerk. Ebenfalls auf anschlußfertige Diskettenlaufwerke hat sich Concorde spezialisiert: 5¼-Zoll-Floppy-Disk-Laufwerke ab 163 KByte Speicherkapazität (235 Dollar) für Apple II, IBM-PC/XT, TRS-80, TI 99/4A, Commodore- und Atari-Heimcomputer.

Beliebt sind Lichtgriffel: Am interessantesten ist aufgrund der mitgelieferten Software wohl der Gibson Lightpen für Apple II, IBM-PC, PC junior und Commodore-Computer. Preis: um die 300 Dollar. Die Software, die mit diesem Lichtgriffel arbeitet, wird ähnlich der sein, die es für das Koala-Pad — das ein Renner unter den Eingabegeräten für Heimcomputer zu werden scheint — gibt. Für den Commodore 64 bietet Inkwell Systems das Grafikprogramm Flexidraw mit Lichtgriffel an. Interessant: Das Zusatzprogramm Penpal erlaubt das Übertragen von Bildern zu anderen Commodore-64-Computern über Modem — und zwar in dem Moment in dem das Bild entsteht.

Billigst-Thermodrucker bietet Alphacom an: 99,95 Dollar kostet der Alphacom 42 (40 Zeichen pro Zeile) inklusive Kabel für Atari- und Commodore-Computer. 169,95 Dollar sind für den Alphacom 81 (80 Zeichen pro Zeile) zu zahlen, zusätzlich 44,95 Dollar für ein Druckerkabel. (sc)



Wir wollen in dieser Rubrik regelmäßig Bücher vorstellen. Dabei sollen nicht nur die neuesten Bücher besprochen werden, sondern auch Bände, die bereits länger auf dem Markt sind, finden Eingang. Dadurch erhalten Sie mit der Zeit eine relativ vollständige Liste aller Veröffentlichungen, die den 64 und VC 20 betreffen.

Sie besitzen sicherlich bereits eine ganze Menge an Büchern über den VC 20 und den Commodore 64. Haben Sie sich dabei über ein Buch besonders gefreut oder geärgert, so lassen Sie uns doch Ihre Meinung zukommen. Wir sind dankbar für jeden Beitrag, der dem Leser hilft, sich für das richtige Buch zu entscheiden.

Betriebswirtschaft auf dem 64

Wer behauptet, der Commodore 64 sei betriebswirtschaftlich nicht zu nutzen, wird durch das Buch »Wirtschaft auf dem Commodore 64« von J. Elsing und D. Herrmann, IWT-Verlag GmbH, Vaterstetten bei München, ISBN 3-88322-030-2, für 38 Mark, eines besseren belehrt.

Fragestellungen der Finanzmathematik, der Unternehmensforschung (Operations Research) und der Betriebswirtschaft werden weitgehend aufgezeigt, und es werden durch Programme Lösungswege angeboten. Von der einfachen Zinsrechnung über die Einkommens- und Lohnsteuerberechnung bis hin zur Zeitreihenanalyse ist in diesem Buch alles Wesentliche vertreten.

In sachlich nüchterner Form werden die einzelnen Berechnungsarten vorgestellt. Nach einer kurzen Einleitung werden dann die angewandten Formeln erläutert, die in den Programmen benutzt werden. Situationsbeschreibungen, die aufzeigen, für welche Problemlösungen die einzelnen Programme genutzt werden können, schließen sich an. Am Ende der kurzgehaltenen Kapitel sind dann die Programme selbst abgedruckt. Jede Berechnungsart wird einzeln durch ein Programm abgearbeitet. Für den Commodore 64-Besitzer dürfte es allerdings keine Schwierigkeit sein, die Programme zusammenzufassen und durch ein Menü dann abzurufen. Hervorzuheben sei noch, daß einige der Programme auch die guten grafischen Darstellungsmöglichkeiten des Commodore 64 ausnutzen. Die Grafik-Programmteile sind so angelegt, daß sie bei Bedarf leicht entfernt werden können, doch warum sollte man auf diese Möglichkeit der grafischen Auswertung verzichten.

Der betriebswirtschaftliche Anwender wird kaum auf dieses Buch und seine Programme verzichten wollen. Dazu sei noch erwähnt, daß eine das Buch begleitende Diskette, auf der alle Programme enthalten sind, angeboten wird. (rg)

Das Interface Age Systemhandbuch zum 64 und VC 20

Das Buch mit dem obigen Titel vom Interface Age Verlag ISBN 3-88986-001-X, Preis 74,— Mark, 306 Seiten stark, der Autoren Babel, Krause und Dripke bietet eine faszinierende Fülle an Daten für den Commodore 64. In diesem Buch wird das Betriebssystem des Commodore 64 (unter Einbeziehung des VC 20) detailliert dargestellt. Es wendet sich daher primär nicht an den Anfänger, sondern soll den Fortgeschrittenen sowohl in der Basic- als auch der Assembler-Programmierung als Nachschlagewerk dienen. So beginnt das erste Kapitel auch gleich mit der Vorstellung des Basic-Interpreters und mit einigen neuen Tips zur Anwendung der USR-Funktion. Der nächste Teil beschäftigt sich mit der Assembler-Programmierung. Durch die Beschränkung auf 15 Seiten ist dies natürlich kein Ersatz für ein ganzes Buch über Assembler, aber eine notwendige Voraussetzung zur effektiven Ausnützung des im zehnten Abschnitt erscheinenden ROM-Listings (insgesamt 157 Seiten). Eingegangen wird außerdem noch auf die Grafik und Farbe (sprich VIC-II-Chip), auf die Tonerzeugung (SID 6581 Chip), auf die Ein-/Ausgabe, auf die Echtzeituhr im CIA Chip und auf die Speicheraufteilung. Ein besonderer Vorzug dieses Buches ist der häufige Vergleich der Speicheradressen zwischen dem Commodore 64 und dem VC 20 sowie ein Abschnitt über die Adaption von CBM-Programmen an den Commodore 64. Unverständlich hingegen warum ein Buch, das von deutschen Softwarespezialisten geschrieben wurde, keine Umlaute und ß enthält. So sehr sollte man sich dann doch nicht an der amerikanischen Schreibweise des Commodore 64 orientieren, und die deutsche Rechtschreibung verleugnen. Ansonsten ist der Text verständlich gehalten, wenn auch einige Kapitel aufgrund der komprimierten Informationsvermittlung des öfteren zu studieren sind, um die gebündelte Information voll auswerten zu können. (aa)

64 für Profis

Um es gleich vorweg zu nehmen, der Inhalt hält nicht was der Titel verspricht. Dieses Data Becker Buch von den Autoren Angershausen, Becker, Gerits und Schellenberger, 276 Seiten, ISBN 3-89011-007-X für 49 Mark ist nicht für Profis geschrieben. Diese wissen nämlich bereits das meiste, was hier an Informationen für Profis drinsteht (ansonsten würden sie eine derartige Bezeichnung nicht verdienen). An wen richtet sich dann der siebte Band von Data Becker? Ganz klar an den Fortgeschrittenen bei der Anwendungsprogrammierung in Basic. Dieser Fortgeschrittene ist mit seinem Computer, speziell dem Commodore 64, bereits aufs engste vertraut, und kann alle (selbst)gestellten Aufgaben mehr oder weniger richtig lösen. Welchen Nutzen soll der Fortgeschrittene aus diesem Buch ziehen? Er soll sich die Programmiertechniken und Vorgehensweisen eines Profis zueigen machen. Dies sind vor allen Dingen die effiziente Programmerstellung, der Gedanke an den Benutzer und die Änderungsfreundlichkeit — Punkte, die der Hobbyprogrammierer nur zu oft unbeachtet läßt. Effiziente Programmerstellung ist gekennzeichnet durch modularen Aufbau und sorgfältige Planung, im Idealfall kein Renumber nach der Fertigstellung des Programms. Beim Gedanken an den Benutzer sollte das Programm so perfekt sein, daß es nicht unvermutet aussteigt oder der Anwender ohne Bedienerführung am Bildschirm alleingelassen wird.

Fortsetzung auf Seite 139



Wer kennt Tiny-Basic?

Ich habe mir den Tiny-Basic-Compiler von Abacus gekauft, erhielt aber keine Beschreibung. Wer kann mir helfen? Wer kann über Anwendungen berichten?

Carl Becker

Fernschreiber an 64er anschließen?

Wie kann ein Fernschreiber an den Commodore 64 angeschlossen werden? Wer hat entprechende Schaltpläne und ein Programm? Wer verkauft Fernschreiber?

Andreas Wecks

Textverarbeitung mit VC 20?

Ich suche dringend eine solide Einführung in die Textverarbeitung für VC 20 — Buch, Broschüre oder Artikel. Wer kann mir einen Hinweis geben?

Hans Hohenwarter

Maschinenroutinen für Master 64?

Ich möchte bei meinem 64, der unter Master 64 läuft, eigene Routinen in Maschinensprache einbauen. Mit der üblichen Methode (Kassettenpuffer) funktioniert das nicht. Wo lassen sich beim Master 64 solche Maschinenroutinen einbauen?

Günther Henck

Wie realisert man den Datenaustausch?

Besteht die Möglichkeit des Datenaustausches zwischen zwei Commodore 64? Wer hat ein Programm dafür?

Jörg Hesse

Ich besitze einen Commodore 720 mit Floppy-Laufwerk 8250. Gibt es eine Möglichkeit, den Commodore 64 mit dieser Anlage zu koppeln?

Andreas Degenhart



Wer hat Bauanleitungen?

Wo kann man Bauanleitungen für verschiedene Peripheriegeräte wie Drucker, Plotter, Modem etc. erhalten?

Ralph King

Ich möchte meinen Commodore 64 mit einem IEEE-488-Interface ausrüsten. Wer kann mir dazu einen Schaltplan oder einen Bausatz anbieten. Da ich einen Drucker cbm 4022 besitze und diesen an den Commodore 64 anschließen möchte, benötige ich dringend dieses Interface, aber möglichst im Selbstbau.

Reinhard Gervelmeyer

Ich suchen einen EPROM-Burner für den 64. Wer hat dafür eine Bauanleitung?

Matthias Walczyk

Schaufensterwerbung mit Computer?

Wir sind seit kurzem Besitzer eines Commodore 64 mit zwei Floppylaufwerken. Um der Laufkundschaft aktuelle Informationen aus dem Kapitalmarkt und dem Immobilienmarkt geben zu können, möchten wir gerne einen Monitor in unserem Schaufenster installieren und über diesen die Information mitteilen, wobei die Einheit in der oberen Büroetage verbleibt. Besteht die Möglichkeit, die Daten auf Diskette zu speichern und permanent abzuspielen? Gibt es bestimmte Programme für solche Anwendungen?

Ulrich Schipporeit

Dia-Show für 64?

Für den Commodore 64 gibt es ein Programm »Dia Show II«. Diese Dia-Show enthält Bilder in hochauflösender Grafik. Die Bilder stammen vom Apple II und wurden laut Programm auf den Commodore 64 vom Apple II übertragen. Wie kann ich meine Bilder vom Apple auf den Commodore 64 übertragen (64 + Disk VC 1541)? Wie lade ich solche Bilder in den Rechner?

Detlef Wacker

Fragen Sie doch!

Selbst bei sorgfältiger Lektüre von Handbüchern und Programmbeschreibungen bleiben beim Anwender immer wieder Fragen offen. Viel mehr Fragen ergeben sich bei Computer-Interessenten, die noch keine festen Kontakte zu Händlern, Herstellern oder Computerclubs haben. Sie können der Redaktion Ihre Fragen schreiben oder Probleme schildern (am einfachsten auf der beigehefteten Karte). Wir veranlassen, daß die Fragen von einem Fachmann beantwortet werden. Allgemein interessierende Fragen und Antworten werden veröffentlicht.

Multidata 64 kopieren?

Für den Commodore 64 habe ich von Commodore die Standard-Software »Multidata 64« gekauft. Da ich täglich mit der Diskette arbeite, habe ich das Bedürfnis nach einer Sicherheitskopie. Das Kopieren gelang mir indessen einfach nicht. Ich besitze zwei Laufwerke VC 1541. Können sie mir helfen?

Pierre Düby

Wie erweitert man den VC 20-Speicher?

Ich möchte mir für meinen VC 20 eine Speichererweiterungsplatine mit zwei Steckplätzen bauen (für 1 x 3 KByte und 1 x 16 KByte). Dabei möchte ich wie folgt schalten können: a) nur Grundversion, b) Grundversion + 3 K, c) Grundversion + 16 K. Einzelne Bereiche in den Erweiterungsmodulen brauchen nicht schaltbar sein. Es soll nur das lästige Ein- und Ausstecken vermieden werden. Jetzt zu meiner Frage: Reicht es aus, wenn ich nur die Leiterbahnen jeweils zu Pin 21 (+5 V) mit einem Schalter unterbreche, damit dann die jeweils abgeschaltete Erweiterung komplett außer Betrieb ist? Oder kann es durch die anderen nicht unterbrochenen Pins noch zu Störungen oder Beeinflussung kommen? Ich besitze die Original-Commodore-Erweiterungen.

Ludger Kappen

Wer kennt den Elcomp-Wordprozessor?

Hat schon jemand den Elcomp-Wordprozessor aus Hofackers Buch »Programme für den VC 20« zum Laufen gebracht oder ist da ein Fehler im Programm?

Horst Girschick



Nicht immer nur Basic

Ich habe mir für meinen Commodore 64 das Forth 64-Modul von Datatronic AB gekauft. Leider ist es unmöglich, damit zu arbeiten, da die Handhabung des Bildschirmeditors, des Assemblers und der Disk im Handbuch nicht besprochen wird. Auch Hinweise in diversen Forth-Lehrbüchern brachten mich nicht weiter. Vielleicht können Sie mir helfen?

Siegfried Schwarze

Ich suche ein leistungsfähiges Pascal für den Commodore 64. Gibt es außer dem Pascal 64 von Data Becker auch andere Versionen?

Helmut Geiyer

Pascal-Compiler für den Commodore 64 gibt es beispielsweise von Interface-Age, Vohburgerstr. 1, 8000 München 21 und von PHS/SLS, Davenstedterstr. 8, 3000 Hannover 91

Ich besitze einen Pascal-Compiler von Abacus Software für Commodore 64. Wo gibt es eine Bedienungsanleitung oder ein Benutzerhandbuch dafür?

Werner Pfeil

Gibt es für den Commodore 64 einen Fortran-Compiler?

Dirk Nieder

Wer weiß, ob es die Programmiersprache C beziehungsweise einen C-Compiler auch für den Commodore 64 gibt?

Andreas Funk

Zehnertastatur für 64?

Gibt es eine Zehnertastatur (Ziffernblock), den man an den Commodore 64 anschließen kann?

Arndt Grass

Eine Zehnertastatur gibt es von der amerikanischen Firma Cardco, 313 Mathewson Wichita, KS 67214, USA.



64 KByte RAM für VC 20?

Für den VC 20 werden auch 64-KByte-RAM-Erweiterungen angeboten. Lohnt sich der Kauf?

Joachim Grzescik

Ich besitze eine 64-KByte-RAM-Erweiterung. Die Spiele für den VC 20 mit den Erweiterungen 3 K, 8 K oder 16 K laufen bei mir nicht, obwohl die Schalterstellung stimmt. Ist die Erweiterung wertlos?

Michael Dürr

Der VC 20 kann — das ist vom Hersteller vorgesehen — bis knapp 30 KByte RAM aufgerüstet werden. Eine weitere Aufrüstung des RAM ist nicht sinnvoll, da dieser zusätzliche Speicher nicht adressiert werden kann. Uns ist bisher noch kein Fall bekannt geworden, in dem eine 64-KByte-Erweiterung für den VC 20 sinnvoll genutzt worden wäre.



Vergleichstabelle für Speicherbelegung?

Wo finde ich eine Tabelle, die die Speicherbelegung von VC 20 und Commodore 64 gegenüberstellt?

Klaus Russel

»Das Interface-Age-Systemhandbuch zum Commodore 64 und VC 20« erläutert alle Betriebssystem-Unterschiede. Es enthält auch eine Liste der POKE-Befehle sowie einen Vergleich des ROM-Bereichs. Das Buch ist bei Commodore-Händlern, im Buchhandel oder bei Distributor-Interface-Age (Vohburger Str. 1, 8000 München 21) erhältlich und kostet 74 Mark.

Statistik mit 64?

Wo gibt es Statistikpakete für den Commodore 64?

Ronald Blachnik

Statistikprogramme, die auf dem Commodore 64 laufen, bieten unter anderem folgende Firmen an: Ebel, Westring 6, 6107 Rheinheim 1; Computerdienst, Weenderlandstr. 3, 3400 Göttingen; Grabowski, Spechtweg 25, 7800 Freiburg; Software 2001, Humboldtstr. 120, 5000 Köln 90.

Wollen Sie antworten?

Wir veröffentlichen auf dieser Seite auch Fragen, die sich nicht ohne weiteres anhand eines guten Archivs oder aufgrund der Sachkunde eines Herstellers beziehungsweise Programmierers beantworten lassen. Das ist vor allem der Fall, wenn es um bestimmte Erfahrungen geht oder um die Suche nach speziellen Programmen beziehungsweise Produkten. Wenn Sie eine Antwort auf eine hier veröffentlichte Frage wissen — oder eine andere bessere Antwort als die hier gelesene — dann schreiben Sie uns doch. Antworten publizieren wir in einer der nächsten Ausgaben. Bei Bedarf stellen wir auch den Kontakt zwischen Lesern her.

Wie steuert man Stellmotoren an?

Ist es möglich, mit einem Commodore 64 und einem Floppylaufwerk Stellmotoren und Relais anzusteuern? Wer kann geeignete Geräte und Software vermitteln?

Gerd Wurster

Von Commodore gibt es eine Relais-Karte. Es werden von unabhängigen Herstellern noch verschiedene andere Produkte angeboten; weitere Hinweise sind aber ohne genauere Angaben des beabsichtigten Verwendungszwecks nicht möglich.

Exbasic für 64?

Gibt es Exbasic Level II für Commodore 64?

K. J. Bos

Exbasic Level II ist für VC 20 und für Commodore 64 erhältlich. Beispielsweise bei Markt & Technik-Buchladen oder bei Interface-Age (Vohburgerstr. 1, 8000 München 21).

CP/M-Software für 64?

Ich bin Besitzer eines Commodore 64 + VC 1541 + CP/M 2.2.-Karte. Frage: Wo gibt es CP/M 2.2.-Software, die angepaßt an den 64er ist, und das richtige Diskettenformat hat? Wie übertrage ich Apple-CP/M 2.2.-Programme zum 64er. Gibt es MBasic und Fortran zu kaufen — wenn ja, wo und wie teuer?

Detlef Wacker

Die einzige uns bisher bekannte Firma, die CP/M-Software für den Commodore 64 anbietet, ist die amerikanische Firma Add On, deren Produkte von der amerikanischen Firma Data 20, 23011 Moulton Parkway, Suite 810, Laguna Hills CA 92653, USA, angeboten werden.

Datasette oder Diskette?

In welchem Umfang kann — ich bin Anfänger — die Datasette ein Diskettenlaufwerk ersetzen?

Hans Georg Walther

Beide sind Massenspeicher — wem das Diskettenlaufwerk zu teuer ist, kann (und muß) ein Kassettenlaufwerk, also die Datasette, nehmen. Das Kassettenlaufwerk ist deutlich langsamer als das Diskettenlaufwerk. Dazu kommt ein zweiter Nachteil: Auf dem Magnetband in der Kassette werden die Daten sequentiell, also der Reihe nach hintereinander gespeichert. Es kommt also erst beispielsweise die Adresse von Herbert Adam, dann die von Erich Berger und so weiter. Wenn Sie die Adressen in der Reihenfolge brauchen, in der sie gespeichert sind, funktioniert das ganz gut; wenn Sie — bei alphabetischer Reihenfolge — erst die Adresse von Weber, dann die von Berger, dann die von Müller und so weiter brauchen, muß das Band immer erst zu der entsprechenden Stelle vorbeziehungsweise zurücklaufen, was äußerst zeitaufwendig ist. Außerdem macht es einige Arbeit, in eine gegebene Reihenfolge später neue Adressen einzufügen. Würde als Speichermedium eine Diskette verwendet, so könnte auf jede Adresse (oder jeden anderen Datensatz) direkt zugegriffen werden (sogenannter Random Access) ohne daß ein Vor- oder Rücklauf des Bandes abgewartet werden muß.

Außerdem müssen Sie auf der Diskette die Daten nicht unbedingt in einer ganz bestimmten Reihenfolge abspeichern (Details sind von der Dateiorganisation und damit der Software abhängig). Faustregel: Mit einem Diskettenlaufwerk arbeitet man schneller und bequemer als mit einem Kassettenlaufwerk. Alle Anwendungen, die sich mit einem Kassettenlaufwerk realisieren lassen, sind auch mit einem Diskettenlaufwerk zu verwirklichen, während es umgekehrt eine Reihe von Anwendungen gibt, die nur mit einem Diskettenlaufwerk sinnvoll zu bewältigen sind.

Hier sind Clubs

Unter dieser Überschrift werden Sie im Leserforum künftig regelmäßig kurze Informationen über Computerclubs, die sich ausschließlich oder zumindest in wesentlichem Umfang mit den kleinsten Commodore-Computern befassen.

Mit VC 20 und 64 befassen sich drei Clubs. Hilfe für Anfänger bieten und kommerzielle Software, beispielsweise für die Fakturierung, entwickeln, will der Computer-Club Nordkirchen, der auch ein monatliches Treffen veranstaltet (Ansprechpartner: Lothar Leitl, Holtweg 22, 4717 Nordkirchen 2, Telefon 02596/12258 oder Uwe Wienand, Kirchstr. 7, 4717 Nordkirchen 3, Telefon 02596/861).

Wir haben in Bonn einen Commodore-64-Anwender Club gegründet. Unser Ziel ist es, den Informationsaustausch über Soft- und Hardware zwischen den einzelnen Anwendern zu verbessern. Deswegen möchten wir ein Club-Info herausgeben und regelmäßige Treffen einrichten. An einen Clubbeitrag ist — soweit es die Umstände erlauben — nicht gedacht. Unsere Kontaktadresse: Im Gries 15, 5300 Bonn 2.

Thorsten Richter

Seit Dezember 1982 existiert der VCAC, der einen Jahresbeitrag von 25 Mark erhebt und auch sogar eine eigene Zeitschrift VC-Data herausgibt. Kontaktanschrift: Jürgen Wagner, Auf der Wiedigsbreite 14, 3500 Kassel.

Software unter den Mitgliedern zu tauschen, ist Ziel der VC 20/cbm-Interessengemeinschaft (Kontaktadresse: Klaus Dieter Keller, Ortsstr. 77, 6650 Bad Homburg 8) und des Computer-Clubs Saarbrücken (Kontaktadresse: Ralf Deibel, Provinzialstr. 139, 6604 Fechingen). Der Saarbrücker Club befaßt sich mit dem VC 20 und mit anderen kleinen Systemen. Die Mitglieder beider Clubs treffen sich regelmäßig einmal im Monat.

In Bruchsal wurde ein Commodore 64-Club gegründet, der unter anderem dem Informationsaustausch mit anderen Benutzerclubs dienen soll. Kontaktanschrift: Steinackerstr. 12, 7520 Bruchsal

Torsten Zimmermann

Seit Anfang Januar 1983 existiert die Commodore 64 User Group Essen. Mitmachen kann jeder, der über einen Commodore 64 verfügt. Das Tätigkeitsgebiet der C 64 U.G.E. reicht von Spielprogrammen bis hin zu mathematischen Programmen. Geplant sind: Entwicklung neuer Betriebssysteme (zum Beispiel Pascal oder Forth), eigener Hardwarezusätze und eines Assemblers. Interessenten wenden sich an Stefan Ullmann, Meistersingerstr. 66, 4300 Essen 13.

Den Raum Wilhelmshaven/Oldenburg sieht der IBS-Computer-Club als Einzugsbereich an. Seine Mitglieder befassen sich mit Commodore 64 und CBM 8032. Ansprechpartner sind Bernd-Michael Stejskal, Mellumstr. 20, 2940 Wilhelmshaven, und Jörg-Andreas Stejskal, Otto-Suhr-Str. 22, 2900 Oldenburg.

Dem Informations- und Programmaustausch dient der Computer-Club Comm & Co. den 16 Schüler des Gymnasiums Ebern in Unterfranken gegründet haben. Die Schüler arbeiten mit Commodore-Systemen 40XX. Ansprechpartner: Dietmar Schnitzer, Sandleite 1, 8601 Deusdorf.

Spielregeln

Wir verschicken keine Prospekte oder ähnliche Produktinformationen — die müssen Sie direkt beim Lieferanten des Produktes anfordern; die Anschrift kann bei uns erfragt werden.

Wir können keine Programme umschreiben oder anpassen. Wenn ein Leser ein von uns veröffentlichtes Programm umgeschrieben hat und bereit ist, das Listing abzugeben, können wir einen entsprechenden Hinweis im Leserforum veröffentlichen.

Ob und wann Antworten auf die veröffentlichten Fragen eingehen, läßt sich nicht voraussagen; wir sind nicht in der Lage, Vormerklisten zu führen und einzelne Leser individuell zu informieren, wenn eine Antwort eingegangen ist. Wir sind aber gern bereit, den Kontakt zwischen verschiedenen Lesern herzustellen, die am gleichen Thema interessiert sind.

Ansprechpartner für den VC 20/Commodore 64-Club in Kaiserslautern ist Mathias Stoffel, Humboldstr. 22, 6750 Kaiserslautern, Telefon 0631/12676.

Ausschließlich auf den Commodore 64 hat sich der Anwender Club München spezialisiert. Mitglieder-Treffen und Sammelbestellungen werden organisiert, Schulungen und Seminare abgehalten, Soft- und Hardware entwickelt und vertrieben sowie eine Club-Zeitschrift herausgegeben. Der Mitgliedsbeitrag beträgt 5 Mark je Monat plus eine einmalige Aufnahmegebühr von 7,50 Mark (für Berufstätige jeweils das doppelte). Kontaktadresse, Justus Erb (C64-ACM), Theresienhöhe 6b, 8000 München 2, Tel. 089/5023659.

Seit einem halben Jahr besteht der erste Computer-Club Untermain. Kontaktadresse: Ulrich Sauer, Danziger Str. 16, 8754 Großostheim 2. Die Aufnahmegebühr beträgt 30 Mark, der Jahresbeitrag 60 Mark. Die Mitglieder, die überwiegend Commodore 64 benutzen, treffen sich Donnerstag abends im Bürgerhaus Aschaffenburg.

Ulrich Sauer



Seit Monaten geistert das Gerücht um die Z80-Karte für den Commodore 64 umher. Was hat es mit diesem Modul auf sich? Lohnt sich die Anschaffung zu diesem Zeitpunkt und stehen einem dann wirklich alle CP/M-Programme zur Verfügung? Lesen Sie einen ersten Erfahrungsbericht.

In der Erwartung, sich ein seriöseres Software-Angebot für den 64er von Commodore zu erschließen, indem man sich nun ein CP/M-Modul beschafft, kaufte ich mir eine Z80-Karte. Schaut man sich nun genauer an, was man sich denn so zugelegt hat, ist man enttäuscht — zu Recht. Das Modul besteht an sich nur aus einer Platine mit einer normalen Z80-CPU und mehreren Treiberbausteinen. Der Systemtakt von 0.984 MHz, der aus der C 64-Hauptplatine stammt, ist etwas dürftig. Die Verarbeitung ist ebenfalls nicht besonders gut. Bei dem getesteten Modul war zum Beispiel bei mehreren Lötpunkten zu wenig Lötzinn vorhanden — das kann durchaus zu Ausfällen führen.

Das Ganze ist in einem schwarzen Kunststoffgehäuse von wenig vertrauenserweckender Form und Stabilität verpackt.

Mit dabei ist ebenfalls die Systemdiskette. Möchte man nun mit CP/M arbeiten, legt man in das Laufwerk die CP/M-Diskette ein, und lädt das »Ladeprogramm« für das BIOS 65 ein. Danach starte man das Programm mit »RUN«. Und nun kommt der enttäuschende Part 2. Denn man muß geduldig sein. Zu dem bekannterweise »langsamen« Laufwerk addiere man die niedrige Taktfrequenz des Z80 und man erhält eine Wartezeit von zirka einer Minute. Bei jedem B-DOS-ERROR oder anderen Fehlern wiederholt sich diese Wartezeit. Wir wollen einmal darüber hingwegsehen und uns das CP/M genauer betrachten. Zuerst muß man feststellen, unter CP/M verfügt man über 44 KByte (48 KByte mit MCVCPM*.*) unter der Voraussetzung, kein Programm im Speicher zu haben. Man muß sich nun vor Augen halten, daß das nicht viel ist, denn man lädt ja noch die eigentlichen Program-

ote
rungen
dem
MODUL



me ein. Die Diskettenkapazität schrumpft ebenfalls auf 136 KByte. Zu dem Diskettenformat muß man noch sagen, es ist Commodore-eigenes Format — Apple II CP/M-Software beispielsweise läßt sich nicht lesen. Die Werte des IOBYTE ASSIGN sind beim Commodore 64 nicht zu ändern.

Außerdem verlangen die meisten CP/M-Programme 80 Zeichen pro Zeile. Diesen Makel kann man a) mit einem Hardware-Zusatz, b) per Software beheben. Die Hardware allerdings ist sehr teuer, man benötigt wiederum einen Monitor. Die Software hingegen generiert 3 x 8 Zeichenmatrix-Zeichen, man muß c) entweder ziemlich weit vom Bildschirm entfernt sitzen oder d) Phantasie besitzen.

Das Commodore-CP/M besitzt außerdem noch eine kleine Besonderheit: Ein Betrieb mit zwei Laufwerken ist unmöglich (derzeit). Bei PIP B: = A: *.* werden nicht etwa zwei Laufwerke angesprochen. Man lege Diskette »A« in »DRIVE 0« — danach (später) Diskette »B«.

Natürlich sind die genannten Besonderheiten mit MOVCPM und SYSGEN für den »Fachmann« zu ändern (wenn auch mit erheblichem Aufwand). Zugute halten muß man die Tatsache, daß das C 64-CP/M adressenmäßig da liegt, wo es hingehört (ab $0100). Das wird durch einen Adreßoffset von $1000 (dez. 4096) erreicht, die Zero-Page der 6510 CPU wird also nicht zerstört!

Ebenfalls getestet wurde MBASIC V.503, diese Programmiersprache gibt es nicht für das C 64-CP/M. Es wurde durch Rechnerkopplung (Apple-C 64) übertragen und angepaßt.

Commodore läßt die CP/M-Käufer nicht nur literaturmäßig allein, nein, auch mit der Software! Das bedeutet für viele Kunden eine herbe Enttäuschung, denn, wie gesagt, das Diskettenformat der VC 1541 ist nicht kompatibel mit Apple, TRS 80, ... eben nur Commodore-kompatibel. Auf Anfrage an Händler, wie es denn mit Commodore-CP/M-Software sei, wurde nur mit den Schultern gezuckt — nicht bekannt. Abschließend läßt sich sagen: das CP/M-Modul ist, betrachtet man den Preis von zirka 200 Mark, eher günstig zu nennen, trotz der oben genannten Nachteile. Allerdings müßte Commodore a) zumindest ein Handbuch mitliefern, b) die CP/M Version (2.2) überarbeiten, und c) die Geschwindigkeit des Computers erhöhen. Dann wäre der Kauf lohnenswert, auch wenn Commodore den Preis erhöhen würde.

(Peter Dassow)

COMMODORE DR UNTER 1000 MARK



Drucker-Testberichte dienen im allgemeinen dazu, dem Leser Neuheiten auf dem Markt vorzustellen. Durch Drukker-Vergleichstests versucht man in der Regel, dem potentiellen Käufer die Kaufentscheidung zu erleichtern. Dabei werden dann Geräte ausgewählt, die sich Konkurrenz machen, sei es vom Preis oder von der Leistung her, jedoch meistens von mehreren Herstellern. Was aber liegt dem Anwender näher, als daß er sich zuerst einmal bei dem Hersteller seines Computers orientiert? Der Vorteil liegt auf der Hand: Man kann davon ausgehen, daß Computer und Peripherie eines Herstellers kompatibel sind. Das bedeutet, daß man sich nicht erst großartige Hardwarezusätze wie Interfaces und teure Kabel beschaffen und einbauen lassen muß und daß auch — wie bei den hier getesteten Commodore-Druckern — der gesamte Commodore-Zeichensatz ausgedruckt werden kann.

Commodore bietet Drucker für vielerlei Anwendung und zu (fast) jedem Preis an. Wir beschränken uns hier jedoch in diesem Bericht auf diejenigen bis 1000 Mark. In diesen Bereich fallen folgende Drucker (siehe Bild 1): der VC 1515, der neue MPS 801, der den VC 1525 ablöst, sowie der VC 1526. Eine Ausnahme bildet der Printer/Plotter VC 1520 (Bild 2), und zwar insofern, da er weder ein Matrix-Drucker ist noch ein Papierformat verarbeiten kann wie die anderen. Aufgrund dieser Ausnahmestellung haben wir ihn nicht direkt mit den anderen drei verglichen.

VC 1515, MPS 801 und VC 1526

Gemeinsamkeiten:

Einige wichtige Eigenschaften haben alle diese Drucker gemeinsam: Sie lassen sich problemlos über ein mitgeliefertes Kabel an die serielle Schnittstelle des VC 20/C 64 anschließen (Bild 3). Sie verstehen den gesamten Commodore-Zeichensatz, wie Steuerzeichen, Grafik-Symbole und Klein-/Großbuchstaben, und sie sind Matrix-Drucker, das heißt sämtliche Zeichen werden aus einzelnen Punkten aufgebaut, im Gegensatz zu Typenrad- und Kugelkopfdruckern, in denen jedes Zeichen »auf einen Schlag« auf das Papier gebracht wird. Die Anzahl der Punkte, aus denen jedes Zeichen

UCKER



Bild 1. Von links nach rechts: VC 1515, MPS 801 und VC 1526

Wer schon einmal ein etwas längeres Programm geschrieben hat, weiß um die Vorzüge eines Druckers, der ihm das lange Listing übersichtlich präsentiert. Wer seinen Computer auch zur Textverarbeitung einsetzen will, braucht sowieso einen Drucker.

Bild 2. Printer/ Plotter VC 1520

besteht, bestimmt die optische Qualität des Schriftbildes. Je mehr Punkte, desto besser ist in der Regel das Schriftbild.

VC 1515: Wenn der VC 1515 anfängt zu drucken, hat man unwillkürlich das Gefühl, sich wegen der enormen Lautstärke entschuldigen zu müssen. Man braucht schon einige starke Nerven, besser noch einen Gehörschutz, um dieses »Geräusch« längere Zeit ertragen zu können. Es gibt jedoch langsam Schwierigkeiten ihn zu erwerben: Commodore-Händler haben ihn bereits aus dem Programm gestrichen. Wegen seiner großen Verbreitung wurde er jedoch dennoch in diese Übersicht mit aufgenommen. Der VC 1515 ist der kleinste der drei Drucker. Und das bezieht sich nicht nur auf die Gehäusegröße, sondern auch auf das Papierformat: Im Gegensatz zu den beiden »größeren Brüdern«, die bis zu 10 Zoll Papier verarbeiten, verarbeitet der 1515 lediglich 8 Zoll breites Papier.

Der Papiereinzug ist — mit etwas Fingerspitzengefühl und nach einigem Probieren — verhältnismäßig einfach und problemlos. Allerdings vermißt man einen Drehknopf an der Seite des Geräts zum manuellen Papiertransport. Lediglich ein schmales Rädchen erleichtert das Einfädeln des Papiers und ermöglicht einen Papiervorschub (siehe Bild 4). Eine andere Möglichkeit ist, das Papier selbst direkt herauszuziehen. Eine automatische Vor-

schubeinrichtung fehlt völlig. Auch ein Zurückziehen oder -drehen ist nicht möglich. Zum Wechseln des Papiers muß man es deshalb hinter dem Drucker abreißen und den noch im Gerät verbleibenden Rest nach oben herausziehen. Ein nicht besonders bedienungsfreundliches Verfahren.

Schmutzige Finger

Mit dem Einlegen des Farbbandes hat man am Anfang seine liebe Not: Da die Kassette nicht aus einem Teil besteht, sondern aus zwei nur durch das Band verbundene Hälften (siehe Bild 5), mag es nicht auf Anhieb gelingen, das Farbband ordnungsgemäß einzulegen. Und sich dabei die Finger nicht schmutzig zu machen, ist auch ein Problem.

Möglichkeiten der Steuerung

Es dürfte klar sein, daß dieser Drucker nicht die Möglichkeiten eines 2000-Mark-Druckers besitzt. So muß man sich mit Normalschrift und einfacher doppeltbreiter Schrift begnügen. Für einfache Anwendungen ist das auch völlig ausreichend — und nur dafür ist der VC 1515 auch konzipiert worden.

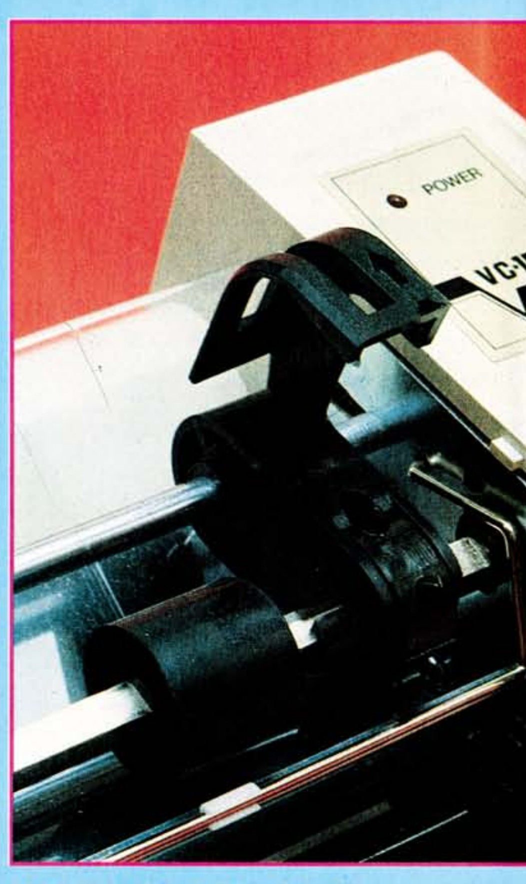
Wie aus der Übersicht zu erkennen ist, werden Zeichen in einer 5 x 7-Matrix dargestellt. Die einzelnen Nadeln des Druckkopfes werden mit der sogenannten Einhammer-Methode aufs Papier geschlagen. Das heißt, daß ein kleiner, mechanischer Hammer jede Nadel einzeln anschlägt und auf das Farbband drückt. Welche Nadel das ist und in welcher Reihenfolge die Nadeln angeschlagen werden, organisiert die Elektronik.

Grafik

Der Grund für die große Anzahl von VC 1515-Besitzern dürfte sicherlich seine Grafikfähigkeit sein. Wie schon erwähnt, ist jedes Zeichen aus einer 5 x 7-Matrix zusammengesetzt, das heißt 5 Punkte breit und 7 Punkte hoch. Da auf einer Zeile 80 Zeichen Platz finden, ergibt sich somit eine Anzahl von 5 x 80 = 400 Punkten in der Breite. Da zwischen jedem Zeichen ein Punkt Abstand ist, erhöht sich die gesamt mögliche Anzahl Punkte auf 480 Spalten pro Zeile und 7 senkrechte Punkte pro Spalte. Jeder dieser Punkte ist einzeln adressier- und druckbar. Das ermöglicht unter anderem den Hardcopy-Ausdruck der hochauflösenden Grafik des VC 20/C 64, aber auch die Gestaltung eigener, individueller Zeichen und komplexer Grafiken.

Bild 3. Serielle Schnittstelle, beim VC 1526 und MPS 801 in doppelter Ausführung





MPS 801: Keine Revolution!

Das erste, was beim Auspacken dieses Druckers auffällt, positiv auffällt, ist das neue Design. Die große und dicke Schallschutzhaube ist der Form des Gerätes sehr gut angepaßt und mindert die Druckerlautstärke auf ein gut erträgliches Maß. Auf komfortable Bedienungs-

COMMODORE DRUCKER IM TEST
ZEITVERGLEICH

Bild 6. Druckzeit (in sec) für: FOR I=1 TO 100: PRINT I,: NEXT I

COMMODORE DRUCKER IM TEST
ZEITVERGLEICH

Bild 7. Druckzeit für: FOR I=1 TO 100: PRINT I : NEXT I

Bild 8. Druckzeit für: FOR I=1 TO 100: PRINT I;: NEXT I

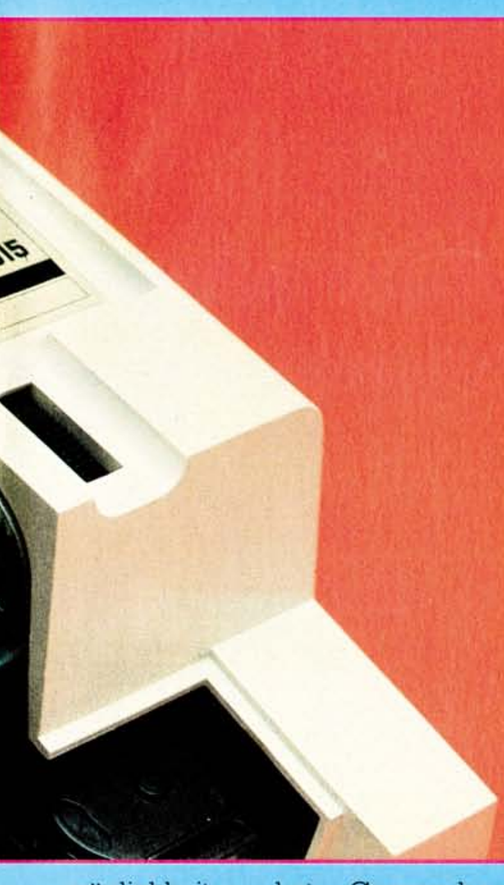


Bild 4. Tractorführung mit hochgestellter Klammer und Rad zum manuellen Papiervorschub beim VC 1515

möglichkeiten hat Commodore (wie bei übrigens allen drei Druckern) anscheinend keinen großen Wert gelegt. So ist auch beim MPS 801 lediglich ein mechanisches Handrad an der rechten Seite des Druckers zum manuellen Papiertransport vorhanden und zusätzlich ein Folienschalter zum schrittweisen automatischen Vorschub.

Die Designer haben aber nicht beim Gehäuse haltgemacht: Auch die Farbbandkassette erfuhr eine erhebliche konstruktive und optische Verbesserung gegenüber den anderen Commodore-Druckern (siehe Bild 5). Die Kassette ist handlich und bedienungsfreundlich. Schmutzige Finger wie beim VC 1515 gibt es nicht mehr und das Einlegen und Wechseln ist ein Kinderspiel. Apropos Wechseln: Wenn die Schrift blasser wird, braucht man nicht mehr das Farbband beziehungsweise die Kassette auszutauschen! Der Clou ist, daß die Kassette einen Tintenbehälter enthält, der leicht durch einen anderen ersetzt werden kann (Bild 6). Eine preiswerte Konstruktion. Bevor der Drucker seine Arbeit aufnimmt, muß das Papier eingezogen werden. Das funktioniert ähnlich wie beim VC 1515 ohne große Komplikationen. Im Gegensatz zu diesem jedoch ist auch ein Zurückziehen des Papiers möglich.

Bild 5. Die Farbbandkassetten der drei Commodore-Drucker: von oben nach unten: VC 1515, MPS 801, VC 1526

Bild 9. Druckzeit für Hardcopy der hochauflösenden Grafik des C 64

Bild 10. Druckzeit für Hardcopy des Bildschirms

Neu — und doch nicht neu

Wenn man allerdings erwartet, einen völlig neuen Drucker vor sich zu haben, wird man spätestens dann enttäuscht, wenn man sich die Möglichkeiten der Druckersteuerung ansieht: An sich ist einem die Sache schon klar, wenn man sich das Handbuch vornimmt (es lag nur in englischer Fassung vor). Wenn nicht der Name MPS 801 mit seinen entsprechenden Bildern da wäre, wäre man überzeugt, die englische Ausgabe des VC 1515-Handbuches vor sich zu haben: die gleiche Aufmachung, die gleichen Beispiele, die gleichen Grafiken und mit den gleichen Druckbefehlen. Man stellt fest, daß sich hier nichts getan hat. Der Unterschied zum VC 1515 liegt dann auch nur im Schriftbild und in der Druckgeschwindigkeit. Zum Schriftbild ist folgendes zu sagen: In der technischen Spezifikation des MPS 801 ist im Feld Zeichenmatrix angegeben: 6 x 7-Punktmatrix (siehe Übersicht), zum VC 1515 steht hierzu: 5 x 7-Punktmatrix. Nun sollte man annehmen, daß der MPS 801 aufgrund dieser besseren Matrix wirklich ein besseres Schriftbild besitzt. Wenn man

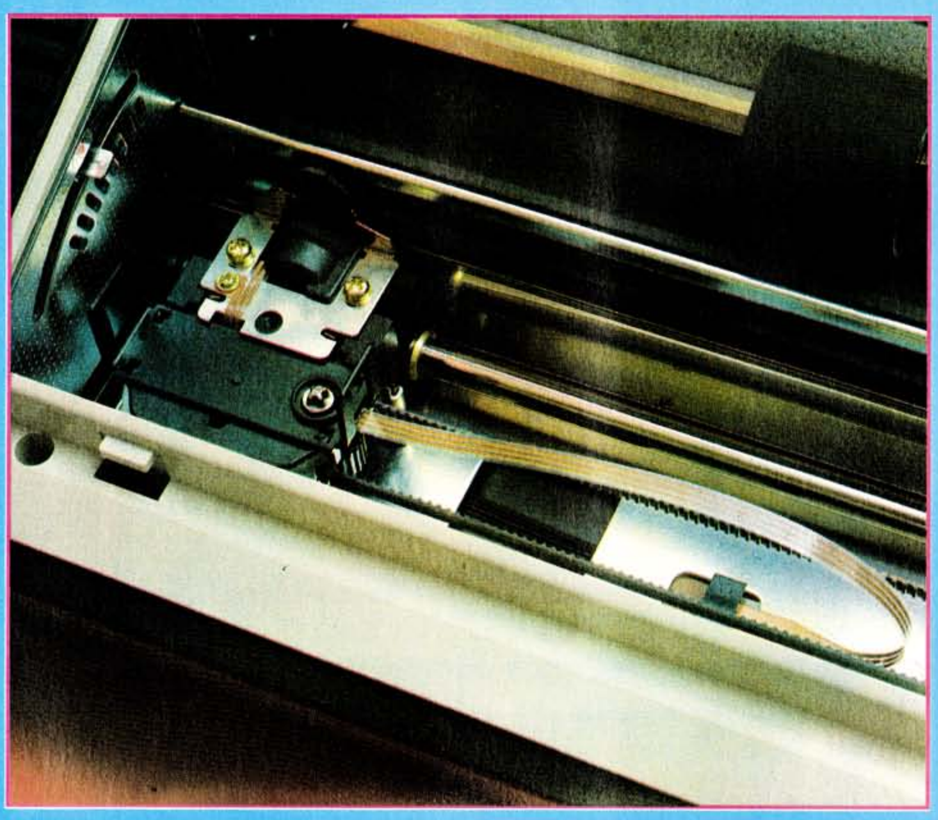


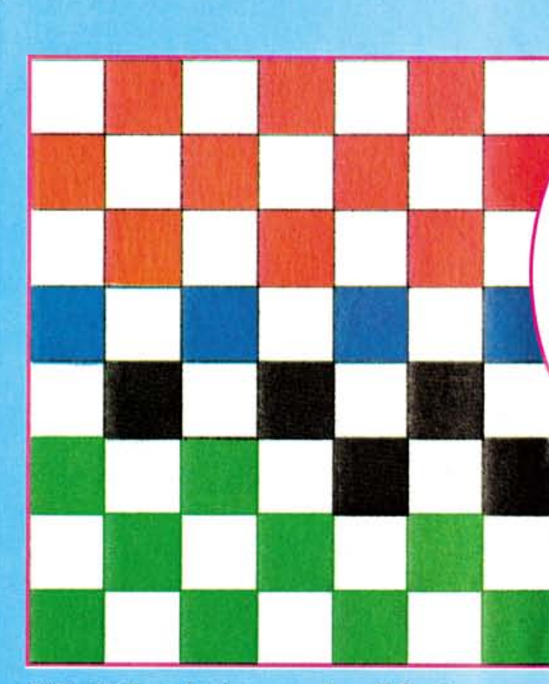
Bild 11. MPS 801 ohne Farbbandkassette. Links vom Druckkopf erkennt man den Hebel zur Anpassung an die Papierdicke



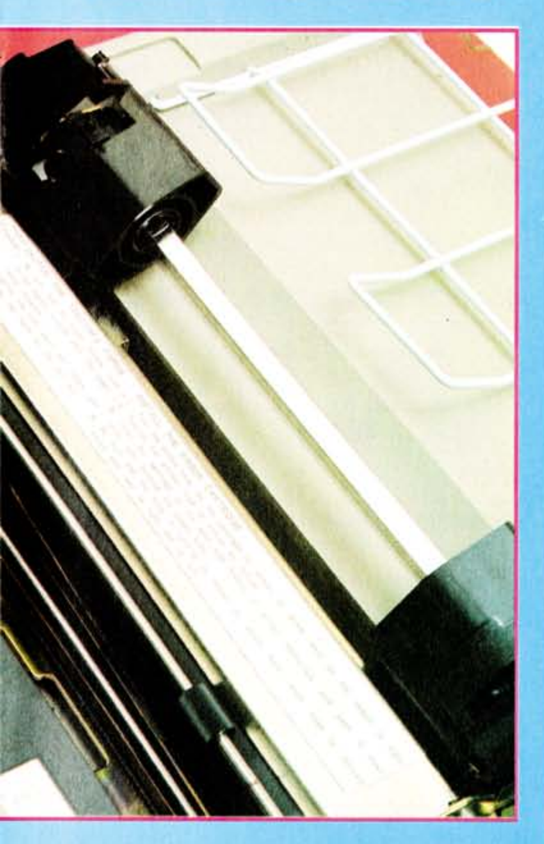
Bild 12. VC 1526 ohne Schallschutzhaube

	VC 1515	*MPS 801*
Druckmethode	Matrix-Druck (Einhammer-Methode)	Matrix-Druck (Einhammer-Methode)
Zeichenmatrix	5x7 Punkt-Matrix	6x7 Punkt-Matrix
Zeichengröße	Höhe: 2,82 mm Breite: 1,76 mm	6x7 Punkt-Matrix
Zeilenlänge	max. 80 Spalten	max. 80 Spalten
Zeilenabstand	6 Z/inch im Zeichenmode 9 Z/inch im Grafikmode	6 Z/inch im Zeichemode, 9 Z/inch im Grafikmode
Druckgeschwindigkeit	30 Zeichen/sek.	50 Zeichen/sek
Druckrichtung	einfache Druckrichtung	einfache Druckrichtung
Papiertransport	Stachelwalzenantrieb	Stachelwalzenantrieb
Farbband-Typ	Kassette 2teilig	Kassette mit auswechselbarem Tintenfaß
Papierbreite	4,5 bis 8 inch	4,5 bis 10 Zoll
Kopien	Original + 2 Kopien	Original + 2 Kopien
Grafik	Punktadressierbar, 7 senkrechte Punkte/Spalte max. 480 Spalten/Zeile	Punktadressierbar, 7 senkrechte Punkte/Spalte max. 480 Spalten/Zeile
Gewicht:	ca. 2,5 kg	ca. 4,8 kg
Preis:	Bei Commodore nicht mehr im Angebot	ca. 795,— Mark

Übersicht. Technische Spezifikationen



sich jedoch die Buchstaben und Zahlen einmal ansieht, stellt man mit Erstaunen fest, daß beide Drucker die gleiche Matrix benutzen, nämlich eine 5 x 7-Matrix! Die Lücke zwischen den Zeichen mit eingerechnet und auch für Grafik-Zeichen benutzt, ergibt sich die 6 x 7-Matrix! Daß das Schriftbild den-



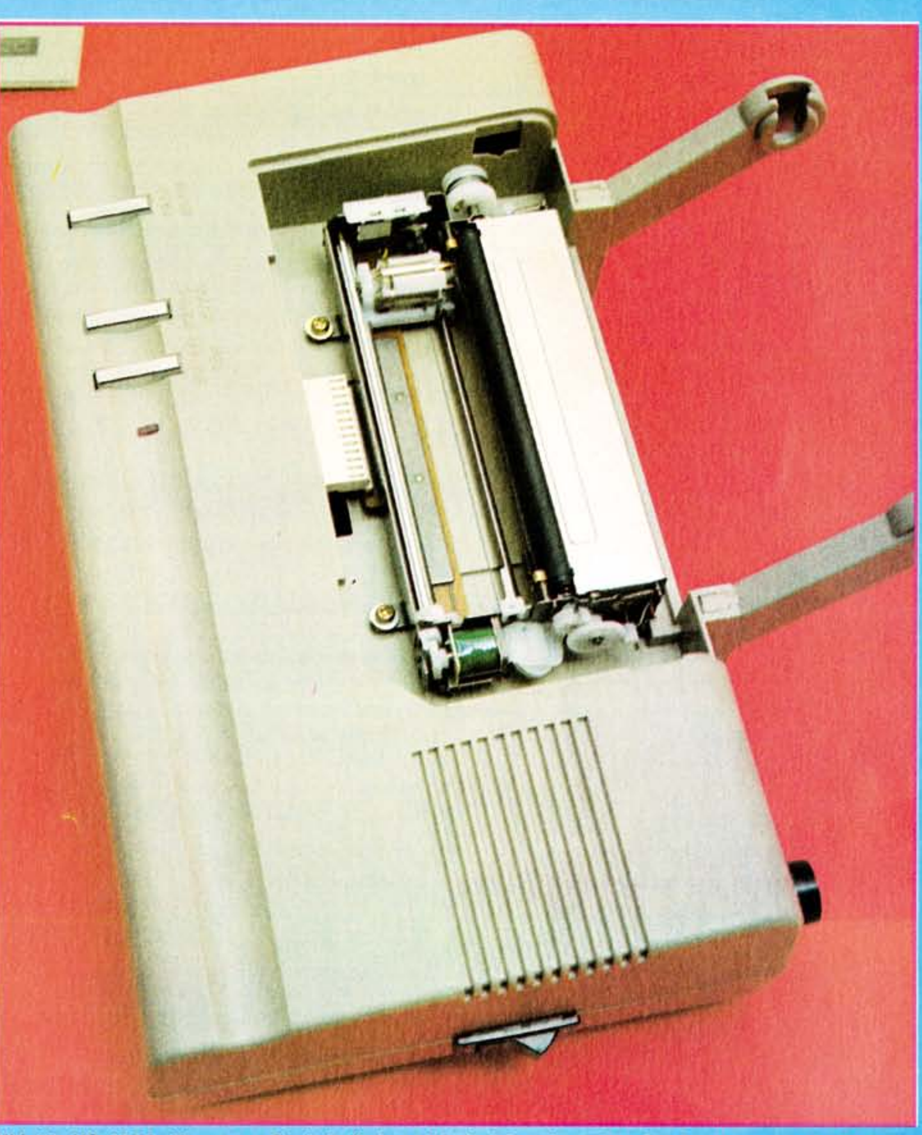


Bild 13. VC 1520. Vorne am Gerät befinden sich drei Druckschalter für Papiervorschub, Farbwechsel und für den Stiftwechsel. Vor der schwarzen Walze sieht man die Trommel mit den kleinen, metallisch glänzenden Farbkugelschreibern. Die beiden Arme an der Rückseite des Plotters nehmen die Papierrolle auf.

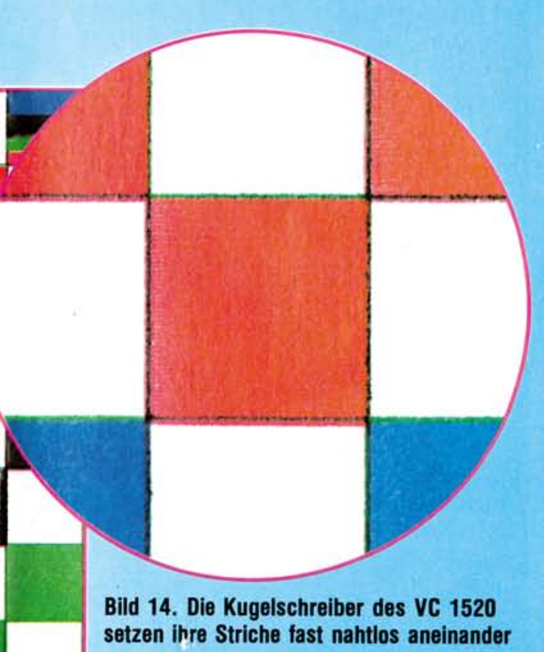


Bild 14. Die Kugelschreiber des VC 1520 setzen ihre Striche fast nahtlos aneinander

	VC 1526	*VC 1520*
Druckmethode	Matrix-Druck (Einhammermethode)	Kugelschreiber
Zeichenmatrix	8x8 Punkt-Matrix	—
Zeichengröße	Höhe: 2,39 mm Breite: 2,03 mm	je nach Schriftgröße
Zeilenlänge	max. 80 Spalten	10, 20, 40 und 80 Zeichen/Zeile
Zeilenabstand	programmierbar	je nach Schriftgröße
Druckgeschwindigkeit	45 Zeilen/min bei 80 Zeichen je Zeile, 78 bei 40 und 124 bei 20	ca. 14 Zeichen pro Sekunde
Druckrichtung	Bidirektional	unidirektional bei Text
Papiertransport	Stachelwalzenantrieb + Gummiwalze f. Einzelblatt	Trommelantrieb, X/P-Plotter
Papierbreite	4,5-10 Zoll	Endlospapier, 4.5 Zoll
Farbbandtyp	Kassette	—
Kopien	Original + 2 Kopien	—
Graphik	nur m. speziellen Programmen möglich	siehe Kasten
Gewicht:	ca. 7 kg	ca. 1,54 kg
Preis:	ca. 895,— Mark	ca. 595,— Markt

Übersicht. Technische Spezifikationen

noch anders aussieht, liegt einerseits an dem breiteren Papierformat des MPS 801 (10 Zoll, die Zeichen werden dementsprechend gespreizt) und andererseits auch an der besseren Qualität des Druckwerkes. Zu den Grafik-Möglichkeiten kann man nur auf den VC 1515 verweisen: Sie sind identisch. Alles in allem ist der MPS 801 keine Revolution. Positiv fällt lediglich das neue Design auf. Laut Commodore bedeutet der Name MPS übrigens ***M***atrix ***P***rinter ***S***ystem.

VC 1526: Der VC 1526 spricht sicherlich eine andere Zielgruppe an als die beiden anderen Drucker. Das bezieht sich weniger auf seinen Preis als auf seine Fähigkeiten. Und da ist er manchen vom Preis her vergleichbaren Druckern in einer Hinsicht überlegen: Seine Formatierungsmöglichkeiten sind ein großer Pluspunkt! Doch dazu später.

Schon rein äußerlich hebt er sich in einigen Punkten von seinen »kleineren Brüdern« ab (Bild 1 und 12). Er ist größer und schwerer. Sein Handrad zum manuellen Papiervorschub sitzt auf der linken Seite und der Ein/Aus-Schalter auf der rechten. Darüber dürften sich wohl alle Linkshänder freuen. Auf der rechten Stirnseite befindet sich ein durchsichtiger Schalter, der, von innen beleuchtet, die Betriebsbereitschaft anzeigt oder durch Flackern einen Fehler erkennen läßt. Durch Drücken des Schalters wird ein Seitenvorschub des Papiers erzielt. Ein zeilenweiser Transport ist allerdings damit nicht möglich, nach meiner Ansicht ein echtes Manko.

Wirklich nachahmenswert und konstruktiv hervorragend gelöst ist der Papiereinzug. Man spannt das Papier (wenn man Lochrandpapier benutzt) auf die Traktorführung und dreht das Handrad. Ein kniffeliges Einfädeln entfällt.

Die Farbbandkassette läßt sich ebenfalls problemlos einlegen. Ein erster Ausdruck läßt ein angenehmes Schriftbild erkennen. Die Großbuchstaben und Zahlen werden in einer 7 x 7-Matrix dargestellt, während Kleinbuchstaben mit Unterlängen die 8. Zeile mitbenutzen. Grafik-Zeichen erreichen dann die volle 8 x 8-Matrix, die auch im Handbuch (siehe Übersicht) angegeben wird.

Stark im Formatieren

Leider besitzt auch der VC 1526 außer der bei den anderen Typen schon erwähnten Breitschrift keine weiteren Möglichkeiten der Zeichendarstellung, wie etwa Fettschrift oder Unterstreichung. Die Zeichen lassen sich jedoch bis zu 3fach verbreitern.

Die wahre Stärke des VC 1526 liegt dann auch in der formatierten Ausgabe von numerischen und alphanumerischen Zeichen. Diese Formatsteuerung ermöglicht eine spaltengerechte Zeichen- und Zahlenausgabe, die Anzahl der Zeilen pro Seite festzulegen sowie Format-Fehlerdiagnose-Nachrichten zu setzen. So kann man zum Beispiel mit wenigen Befehlen festlegen, an welchen Spalten auf dem Papier die Dezimalpunkte der Zahlen stehen sollen, wieviel Nachkommastellen und ob etwa vorlaufende Nullen beziehungsweise ein Dollarzeichen vor der Zahl oder auch ein Vorzeichen mit ausgedruckt werden soll.

Diese Möglichkeiten heben den VC 1526 vom VC 1515 und vom MPS 801 ab und lassen ihm vor allem in Bereichen mit viel Tabellenverarbeitung einen großen Stellenwert zukommen. Unterstützt wird dies noch durch die Möglichkeit, nicht nur Endlospapier, sondern auch Einzelblätter zu benutzen. Nur eines sollte man beim Kauf beachten: Eine Vollgrafik kann man mit dem VC 1526 nur bei spezieller Software bewerkstelligen! Wer oft hochauflösende Grafiken schwarz auf weiß benötigt, sollte diese Einschränkung kennen.

VC 1520 Printer/Plotter	
Farbe	4 Farben (schwarz, blau, grün, rot)
Druckgeschwindigkeit	ca. 14 Zeichen/Sekunde
Zeichengeschwindigkeit	264 Schritte/Sekunde
Zeichenauflösung	0,2 mm (=1 Schritt) entlang der X-Achse 0,2 mm (=1 Schritt) entlang der Y-Achse
Geschwindigkeit	52 mm/s (entlang der X- u. Y-Achse)
beim Ziehen einer Linie	73 mm/s (unter 45 Grad-Winkel)
Zeichenbereich	480 Schritte entlang der X-Achse, entlang der Y-Achse bis +/-999 Schritte programmierbar

Übersicht. Technische Spezifikationen (Schluß)

VC 1520: Der Zeichenspezialist

VC 1520: Von einer ganz anderen Art ist der Printer/Plotter VC 1520 (Bild 2, 7 und 15). Der Name verrät schon: Dieses handliche kleine Gerät ist kein Drucker in dem Sinne wie die anderen. Seine Stärke liegt nicht im Ausdrucken von langen Texten, und er ist auch nicht gedacht als Alternative zu den oben beschriebenen Matrix-Druckern.

Dagegen spricht auch schon sein eingeschränktes Papierformat von 4,5 Zoll. Aber wer einmal die Fähigkeiten des kleinen Zeichners gesehen hat, wird fasziniert sein. Es macht richtig Spaß, mit ihm zu arbeiten, ihm zuzuschauen bei seinen Bewegungen, die er mit seinen vier Farbstiften vollzieht, in jeder Richtung, von links nach rechts, von oben nach unten und umgekehrt. Tatsächlich bewegen sich die Farbstifte dabei natürlich nur in zwei Richtungen, nämlich nur horizontal. Die vertikale Bewegung übernimmt die Gummirolle mit ihren kleinen Nadeln, mit denen sie das Papier blitzschnell vor und zurücklaufen läßt.

Wenn während des Zeichnens einmal die Farbe gewechselt wird (auch manuell durch Tastendruck möglich) und der Zeichenstift wieder genau an der gleichen Stelle weiterzeichnet, wenn die Striche, die der VC 1520 mit seinen vier kleinen Kugelschreibern hinmalt, so eng nebeneinander gesetzt werden, daß man mit bloßem Auge sie nicht mehr trennen kann, dann ist man doch verblüfft (Bild 14). Die Grafiken über den Geschwindigkeitsvergleich aller Drucker stammen übrigens vom VC 1520 (Bilder 6 bis 10). Was allerdings die teureren Plotter meistens eingebaut haben, nämlich eine Software, die das Zeichnen von Skalierungen, Kreisen, Ellipsen, Schraffuren und so weiter mit einfachen Befehlen erlaubt, muß man beim VC 1520 selbst programmieren. Wenn man jedoch systematisch vorgeht, kann man sich schnell die entsprechenden Unterprogramme dafür selbst erstellen und sich so mit der Zeit eine eigene Unterprogramm-Bibliothek aufbauen.

Es ist nicht nur möglich, selbst definierte Grafiken zu erstellen, auch Text läßt sich darstellen, und das sogar in vier verschiedenen Schriftgrößen. Ein Drehen der Zeichen um 90 Grad ist ebenfalls sinnvoll einzusetzen, zum Beispiel zur Beschriftung senkrechter Achsen.

Da der VC 1520 nicht nur ein Plotter ist, sondern eben auch ein Printer, kann man ihn sehr wohl als Ersatz für einen Matrix-Drucker ansehen, mit den schon erwähnten Einschränkungen der geringen Papierbreite und auch der Geschwindigkeit. Wer also keine Korrespondenz führen will beziehungsweise keinen großen Wert legt auf hohe Geschwindigkeit, sondern die Möglichkeiten als Plotter ausnutzen kann, ist mit dem VC 1520 gut bedient.

(gk)

SX 64

im Test



Bild 1.
Der SX 64 aufgebaut

Mobilität ist eine Zauberformel, die unser Leben im letzten Jahrzehnt entscheidend beeinflußt hat. Transistorradio, tragbare Stereoanlage, Fernseher, Mobil-Home, wen wundert's, wenn auch die Computerindustrie, vom Portablefieber erfaßt, mehr und mehr »tragbare« Mini-, Home- und Personal Computer auf den Markt bringt. Vielleicht gehören in naher Zukunft mit Portables bewaffnete »Hacker« am Badestrand unter dem Sonnenschirm genauso zum Strandalltag wie heute Familienväter im Kampf mit dem Gummiboot.

Seitdem 1980 Adam Osborne seinen ebenso viel geschmähten wie hochgelobten Osborne 1 vorstellte und damit die Portable-Lawine ins Rollen kam, erschienen zirka 80 bis 100 »Tragbare« auf dem amerikanischen Markt. In immer neuen Variationen versuchten findige Ingenieure, mehr oder wenig erfolgreich, möglichst viel Hardware auf immer kleinerem Raum und mit immer weniger Gewicht unterzubringen. Auf der Hannovermesse '83 stellte Commodore erstmals seinen lange angekündigten und mit viel Spannung erwarteten Koffercomputer vor, den Commodore Executive als SX 64 mit Single-Floppy beziehungs-

SX 64 im Test



Bild 6. Die Rückseite mit den Anschlüssen

weise DX 64 mit Double-Floppy ausgestattet. Auffallend: der eingebaute Farbmonitor. Wäre nicht das Commodore-Firmenemblem untrüglicher Beweis für die Herkunft des Gerätes, hätte ich vom äußeren Erscheinungsbild her nie auf Commodore getippt. Kein Cremeweiß, keine weichen Rundungen. Nein, stahlgrau, eckig, mit blauem Zierstreifen und modernem Design, so präsentiert sich der SX 64 äußerlich (Bild 1).

Nimmt man den Deckel ab, in dem die Tastatur (Bild 2) untergebracht ist, kommen links der 5-Zoll-Farbmonitor und rechts das querliegende Diskettenlaufwerk (5¼ Zoll à 170 KByte, identisch mit der VC 1541) sowie ein Diskettenablagefach (an dieser Stelle befindet sich beim DX 64 das zweite Laufwerk) zum Vorschein. Rechts daneben eine schmale Klapptüre mit dem Reset-Knopf und sieben Einstellreglern (Bild 3). Hiermit können Lautstärke, Kontrast, Helligkeit, Farbsättigung, Rot-Grünbalance sowie der Bildfang eingestellt werden. Die Einstellung ist stabil, und auch nach mehrmaligem Ein- und Ausschalten des Gerätes mußte ich keine Neueinstellung an den Reglern vornehmen. Gott sei Dank, denn die relativ wackeligen Drehregler konnten mich nicht davon überzeugen, ewig halten zu wollen. Die Module kommen oben in den Steckschacht. Ungeheure Stabilität hingegen strahlt der monströse Tragegriff aus, der gleichzeitig auch als Standfuß dient. Hier versuchte man, so scheint mir, das wettzumachen, was bei der Konstruktion des Computer- und Tastaturgehäuses etwas vernachlässigt wurde, die mechanische Stabilität, die bei ei-



Bild 9. Die Module kommen oben in den Steckschacht



Bild 2. Die Tastatur ist äußerst gelungen: ergonomisch, schön und funktionell.

Bild 5. Dieser Text ist vergrößert dargestellt. Versuchen Sie einmal, ein »O« von einer »0« und diese von einer »8« zu unterscheiden.



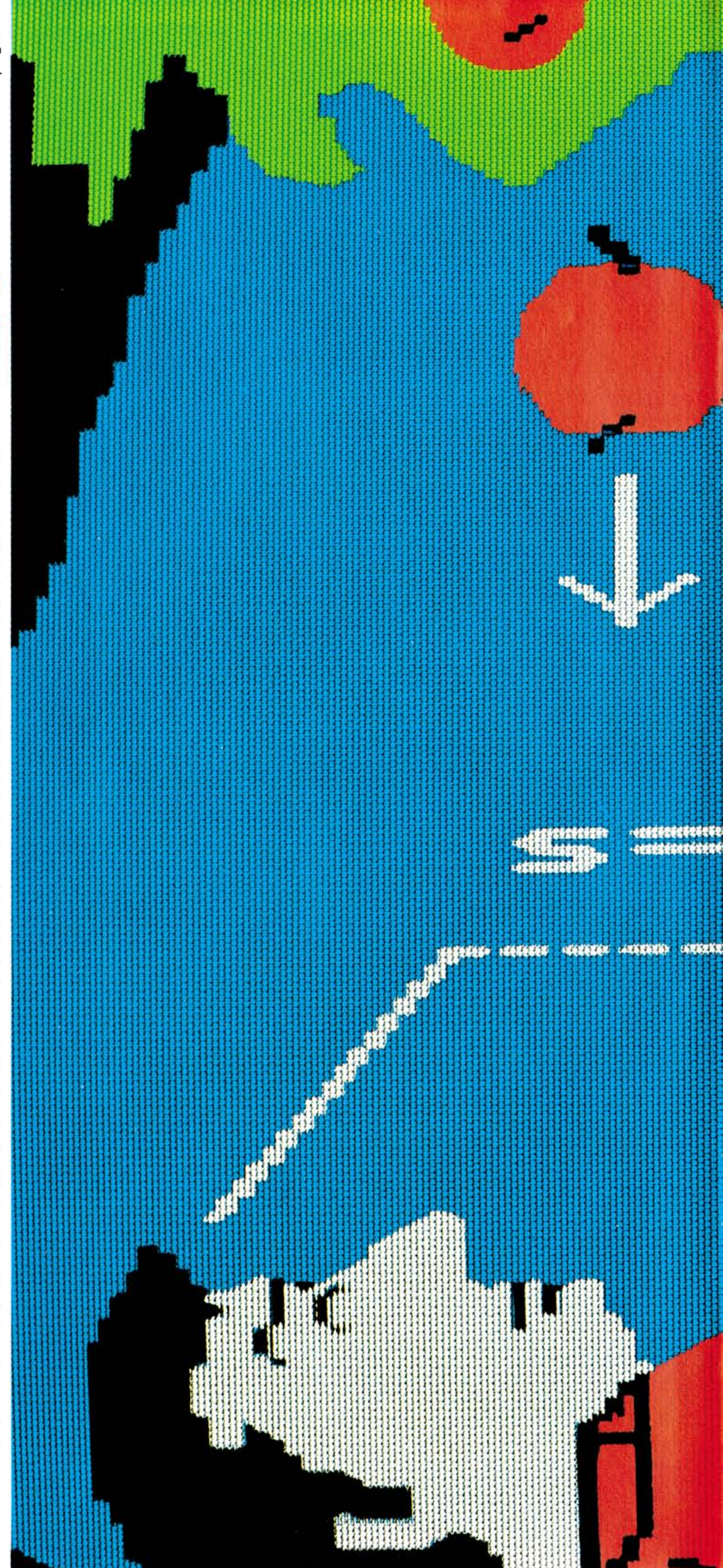
Bild 3. Mit diesen Reglern wird der Bildschirm eingestellt. »Volume« dient zum Variieren der Lautstärke des eingebauten Lautsprechers. Der Resetknopf läßt sich nicht nur mit einem spitzen Gegenstand (etwa Kugelschreiber) auslösen.

nem transportablen Computer sicher eine entscheidende Rolle spielt. So klobig der Griff auch optisch wirkt, so gut liegt er beim Transport in der Hand und läßt zumindest die ersten Kilometer Fußmarsch mit dem SX 64 zu einem Kinderspiel werden. Spätestens nach zehn Minuten jedoch beginnen langsam die Armgelenke zu schmerzen. Man merkt das Gewicht von 10 kg und erkennt, daß sich die Portabilität des SX 64 höchstens auf die Strecken Wohnzimmer — Arbeitsraum oder Wohnung — Garage beschränken wird, soll nicht ein Hanteltraining unumgänglicher Bestandteil des Tagesablaufs werden.

Der große Vorteil des »alles in einem Gehäuse-Gerätes« scheint mir deshalb weniger in der Transportmöglichkeit über längere Strecken zu liegen als in der Tatsache, daß er schnell und ohne Kabelgewirr (das Netzteil ist selbstverständlich eingebaut) betriebsbereit und nach der täglichen Arbeit auch genauso schnell wieder verstaut ist. Mit 5 Zoll Bildschirmdiagonale (13 cm) gestaltet sich die Arbeit jedoch nicht immer zum Vergnügen. Gegen die Farbqualität des Monitors (Bild 4) läßt sich nichts sagen, sie ist hervorragend; ein O von einer Null beziehungsweise die von einer 8 zu unterscheiden, erfordert jedoch viel Einfühlungsvermögen (vergleiche Bild 5). Hier hilft selbst die Brille wenig. Sicher, für die Größe, ser Winzigkeit des Monitors ist die Auflösung ausgezeichnet, aber in diesem Falle wären ein größeres Gehäuse und ein größerer Monitor die bessere Lösung gewesen.

Wer auf dem SX 64 Texte verarbeiten möchte, sollte schon jetzt einen Zusatzmonitor in »Normalgröße« auf den nächsten Weihnachtswunschzettel schreiben, ein Monitoranschluß ist in der Rückseite vorhanden. So entgeht man auch der Gefahr, sich mitten im schönsten Spiel zu zweit vor dem Bildschirm eine Beule am Kopf zu holen bei dem beidseitigen Versuch, noch näher mit den Augen an den Ort des Geschehens zu kommen. Kurz und gut, besten Gewissens kann ich den eingebauten Monitor

SX 64 im Test



nur als Kontrollmonitor empfehlen.

Großes Lob verdienen die 66 Tasten in QWERTY-Anordnung. Die Tastatur stellt gleichzeitig den Deckel des Computers dar. Abgeklappt kann sie, freibeweglich und nur mit einem Verbindungskabel von zirka 50 cm Länge mit dem Gehäuse verbunden, bedient werden. Mit 3 cm Bauhöhe kann man sie im Vergleich zur 8032 SK-Tastatur getrost als für Commodore-Verhältnisse superflach bezeichnen. Commodore vermied Experimente und übernahm das Konzept des vielfach bewährten und beliebten C 64 fast vollständig in den SX 64. Bis auf die ergonomisch bessere Formgebung mit schöner gerundeten Tasten unterscheidet sich die Tastatur weder in Belegung noch Anzahl der Tasten von der des Commodore 64. Das Verbindungskabel Computer/Tastatur erscheint sehr robust. Etwas unpraktisch: Die Steckbuchse an der Unterseite des Tragbaren, die den Kabelstecker aufnimmt, ist in einem Schacht verborgen und dadurch etwas schwer zugänglich. Sehr instabil erscheinen mir die Plastik-Schnappvorrichtungen am Tastaturgehäuse, mit denen dieses am Gehäuse befestigt wird. Sie verklemmten sich bei meinem Gerät nach einem Transport prompt und ich stand alle Ängste aus, die Tastatur nur mit Bruch wieder vom Gehäuse loszubringen.

Auf der Oberseite des Gehäuses ist ein durch Federklappen geschützter Expansionport, das heißt, ein Steckplatz für Module, zum Beispiel das IEEE-488-Interface, Spiele und so weiter. In diesen Steckplatz passen alle für den C 64 bestimmten Module. Über das IEEE-488-Interface ist die gesamte Peripherie der 4000er und 8000er Systeme anschließbar. Ein neues Steckmodul, das in diesen Tagen erhältlich sein soll, und austauschbare Tastenkuppen ermöglichen die Umrüstung der 64-Tastatur auf den deutschen Zeichensatz.

SX 64 im Test

Auf der Rückseite des Gehäuses befinden sich die Peripherieanschlüsse (Bild 6):

☐ Die DIN-Buchse für den Audio- und Videoausgang.

☐ Ein serieller Bus zum Anschluß für das Diskettenlaufwerk VC 1541 und/oder Drucker 1525, MPS 801, VC 1526 beziehungsweise den Plotter VC 1520.

☐ Der Userport als frei programmierbare 8-Bit-parallel-Schnittstelle. Durch entsprechende Programmierung als RS232-Schnittstelle verwendbar.

☐ Zwei Anschlüsse für Joysticks.

Im Inneren des Computers befindet sich die modifizierte Rechnerplatine des C 64, aufgeteilt auf zwei Platinen, sowie die modifizierte Platine der Floppy VC 1541 und ein 8-cm-Lautsprecher, der befriedigende Klangergebnisse erzielt. Die im Gehäuseinneren erzeugte Wärme wird über die Lüftungsschlitze genügend abgeleitet, auch nach einem Dauerbetriebstest von 48 Stunden erwärmte sich der SX 64 nur unwesentlich.



Bild 4. Ein kleines Beispiel für die hochauflösende Grafik

Genau wie der C 64 arbeitet auch der SX 64 mit der 8-Bit-MOS-CPU 6510 aus der Familie 65xx, bei einem Systemtakt von 985248 kHz. Der Speicher verfügt über 64 KByte RAM, wovon in Basic 38 KByte für Programm und Variablen verfügbar sind. 52 KByte können hiervon für den Einsatz von Maschinensprache oder ladbaren Programmiersprachen genutzt werden. In 20 KByte ROM sind das Betriebssystem, der Basic-Interpreter und die I/O-Routinen untergebracht. Da der 6510 als 8-Bit-Prozessor selbst nur einen Adreßraum von 64 KByte verwalten kann, der vom RAM selbst belegt ist, bestand das Kunststück darin, mittels zusätzlicher Logik eine sinnvolle Verwaltung der sich teilweise überlappenden Speicherbereiche auszuklügeln. Hier kam Commodore der glückliche Umstand zugute, über eine eigene Halbleiterfabrikation, nämlich der Tochterfirma MOS zu verfügen. Ein speziell entwickeltes »Adress Manager IC« (FPLA, Field programmable Logic Array) übernimmt diese komplizierte Aufgabe. Auch der Prozessor selbst sowie das Sound-IC, das SID 6581 (ebenfalls ein Peripherie-Baustein der 65xx-Familie) sowie der Videocontroller VIC, der Schlüssel zur hochauflösenden Grafik, gehen auf das Konto der MOS-Entwicklungsingenieure.

Der SX 64 benutzt genau wie der C 64 das Commodore-Basic V 2.0 und ist maschinensprachekompatibel zum 6502. Es können jedoch auch andere Programmiersprachen wie zum Beispiel Pascal, Comal, Pilot, Assembler und Logo geladen werden. Das Basic-ROM wird dann abgeschaltet, und es stehen 20 KByte für die Programmiersprache und den Arbeitsspeicher zur Verfügung.

Das Basic des SX 64 ist identisch mit dem des C 64. Da das V 2.0-Basic in der Literatur bereits zur Genüge abgehandelt wurde, möchte ich an dieser Stelle nicht mehr näher darauf eingehen.

SX 64-Einsteiger brauchen sich über ein mangelndes Angebot an Software keine Gedanken machen, der C 64 hat hier Basisarbeit geleistet. Auch Literatur existiert mittlerweile in Hülle und Fülle. Ohne diese kommt der ernsthafte SX 64-User sowieso nicht aus. Das Bedienungshandbuch ist im Vergleich zum C 64-Handbuch zwar sehr ausführlich, doch viele wichtige Dinge bleiben auch hier wieder unerwähnt oder werden nur dürftig am Rande behandelt. Unverständlicherweise gerade die Bereiche, die den SX 64 interessant machen, nämlich die Erzeugung von Sprites sowie die Möglichkeiten der hochauflösenden Grafik und der Klangerzeugung mit dem SID 6581. Vergebens suchte ich im englischen Handbuch, das mir vorlag, nach dem Befehl, der in den hochauflösenden Grafik-Mode führt. Auch die interessantesten Möglichkeiten des wirklich hervorragenden SID-Chips, nämlich Ringmodulation, Synchronisation und Filterung bleiben gänzlich unerwähnt.

Im hochauflösenden Grafikmodus können 64000 (320 x 200) einzelne Bildschirmpunkte (Pixels) angesprochen werden. Nach dem Einschalten durch POKE 53265,59 : POKE 53272,24 können die einzelnen Bildschirmpunkte mittels POKE x,y gesetzt und mittels POKE x,0 wieder gelöscht werden. Jeder Adresse entspricht hierbei eine Zeile von acht Bildschirmpunkten.

Je nachdem, welche Punkte nun gesetzt werden sollen, setzt man den zugehörigen n-Wert in nachfolgender Formel gleich Null und bildet die Summe.



Bild 8. Diese Farben lassen sich auf dem Bildschirm darstellen

Formel: $y=n\uparrow 7 + n\uparrow 6 + n\uparrow 5 + n\uparrow 4 + n\uparrow 3 + n\uparrow 2 + n\uparrow 1 + n\uparrow 0$ wobei nun der Summenwert y den zu pokenden Wert darstellt. Ein kleines Beispiel soll dies verdeutlichen (Bild 7):

Diese drei Punkte lassen sich mittels POKE 8192, 128+16+8, also POKE 8192,152 setzen. So einfach ist das also. Im Blockgrafik-Modus stellt der Bildschirmspeicher (Adressen 1024 bis 2023) 25 Zeilen und 40 Spalten in einer 8 x 8-Punktematrix zur Verfügung.

Für Farbe im tristen Alltag sorgt der Farbspeicher ebenfalls mit 1000 Bildpunkten (Adressen 55296 bis 56295). An Farben stehen Schwarz, Weiß, Rot, Türkis, Violett, Grün, Blau, Gelb, Orange, Braun, Hellrot, drei verschiedene Grauwerte, Hellgrün und Hellgrau zur Auswahl (Bild 8). Die tausend Farbpunkte stellen den inneren Bildschirmbereich dar. Darüber hinaus existiert noch ein zweiter Bildschirmbereich, der Rahmen, der unabhängig vom inneren Bereich mit denselben 15 Farben eingefärbt werden kann.

Bewegung ins Bild bringen die vom Benutzer frei definierbaren Sprites, Figuren in hochauflösender Grafik, maximal 24 x 21 Punkte groß, die über POKE-Befehle erstellt werden. Maximal acht Sprites dürfen gleichzeitig auf dem Bildschirm bewegt werden. Klänge in den Raum posaunt der SX 64 mit Hilfe des SID 6581, eines kompletten dreistimmigen Synthesizers mit drei Wellenformen (Dreieck, Sägezahn und Pulswelle) je Stimme. Drei Hüllkurvengeneratoren regeln für jede Stimme einen separaten Lautstärkeverlauf der Töne. Rauschgenerator, Filter, Ringmodulator, das, wovon manch großer Synthesizer träumt, ist vorhanden. So verwundert es nicht, daß in jüngster Zeit immer mehr Musiksoftware angeboten wird, die den C 64 beziehungsweise SX 64 in ein »Musikinstrument« mit vielfältigen Möglichkeiten verwandeln.

Fazit

Interessant ist der SX 64 für alle, die viel unterwegs sind und ohne Computer nicht auskommen wollen oder aber ihren Computer auch zu Hause oft auf- und abbauen müssen. Leider besitzt der SX 64 keinen Akkuanschluß, so daß er eigentlich kein Portable im wahrsten Sinne des Wortes ist. Die Schwachstelle am Ganzen: der Bildschirm. Eine Nummer größer wäre in diesem Falle sicher besser gewesen, dafür hätte wohl jeder ein etwas größeres Gehäuse in Kauf genommen. Besonderes Lob verdienen die Tastatur, die ein ermüdungsfreies Arbeiten auch über einen längeren Zeitraum ermöglicht, und das ansprechende Design des Gehäuses. Ein weiteres großes Plus: die völlige Kompatibilität zum C 64 (die Programmodule werden oben eingesteckt; siehe Bild 9) sowie die Möglichkeit, nach Einbau der CP/M-Karte auf das große Angebot an CP/M-Software zurückgreifen können, auch wenn bisher nur wenige Programme, die unter CP/M laufen, auf das Commodore-Diskettenformat umgeschrieben wurden.

(Richard Aicher/aa)

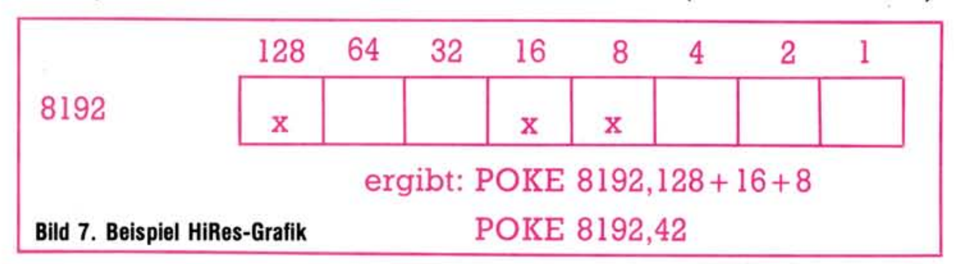


Bild 7. Beispiel HiRes-Grafik

Expansions

Expansions oder auf deutsch gesagt, Erweiterungen, werden dann wünschenswert, wenn man an die technischen Grenzen seines Computers stößt, sei es, daß man zu wenig Speicherplatz hat, oder daß wichtige Verbindungen zur »Außenwelt« fehlen.

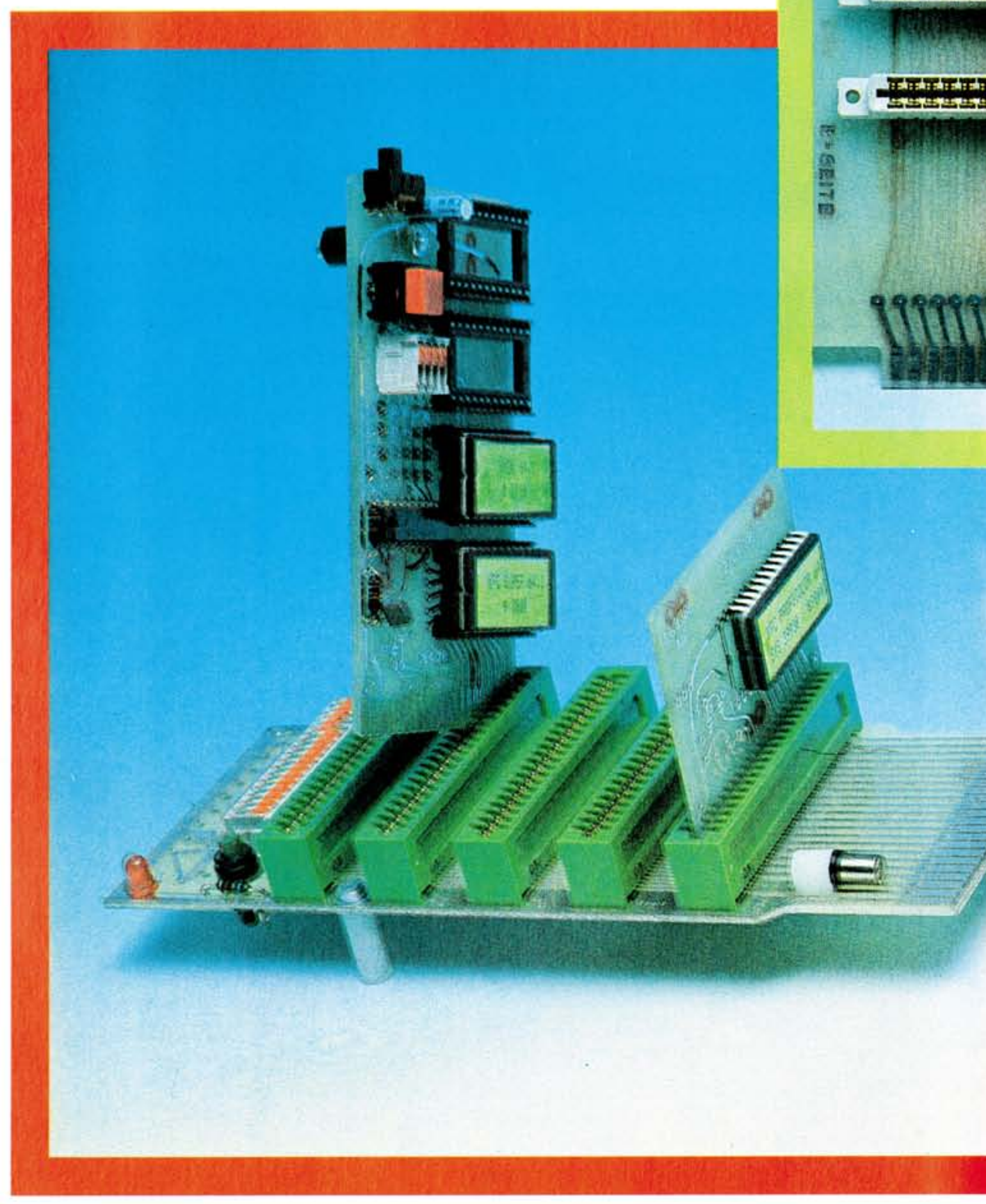


Bild 1. Erweiterungsplatine für insgesamt fünf Module für den Commodore 64. Für drei Steckplätze können Steuersignale des C 64 einzeln zugeschaltet werden, auch ein Resetschalter ist vorhanden (KFC). Das rechte Modul ist ein Graphik-Modul das unter anderem eine Darstellung von 80 Zeichen pro Zeile auf dem Bildschirm erlaubt (KFC), das linke, größere Modul ist das KFC-Super, ein mit EPROMs (das sind Erasable Programmable Read Only Memory = löschbarer programmierbarer Nurlesespeicher (ROM)) erweiterbares Steckmodul.

Waren am Anfang der VC 20/C 64-»Ära« lediglich kleine RAM-Erweiterungen für den VC 20 verfügbar, mauserte sich dieser Markt mit dem wachsenden Erfolg dieser beiden Commodore-Computer. Wir wollen Ihnen regelmäßig auf dem deutschen Markt vorhandene Erweiterungen vorstellen. Sie können sich an dieser Rubrik aktiv beteiligen, indem Sie uns und den Lesern Erfahrungen mitteilen, die Sie gemacht haben. Wenn Sie auf interessante Produkte stoßen sollten, schreiben Sie uns. Auch Anbieter von Erweiterungen bitten wir, uns Informationen zukommen zu lassen. Davon profitieren nicht nur Sie, sondern auch unsere Leser.

Man kann die Expansions ganz grob in vier Themen aufteilen:

Da sind einmal die Schnittstellen (oder Interfaces). Sie ermöglichen eine Verbindung mit Peripheriegeräten, die nicht die serielle Schnittstelle des VC 20/C 64 besitzen.

Zum anderen sind da die Modul- und Steckboxen, die den Einschub von Spielmodulen, Speichererweiterungen und auch Spracherweiterungen erlauben.

Drittens gibt es Erweiterungen, die das Handicap des C 64, aber vor allem des VC 20, der Anzahl von 40 beziehungsweise von 22 Zeichen pro Zeile beseitigen, indem sie (mit Hilfe eines Monitors) eine 80-Zeichen-Ausgabe auf dem Bildschirm ermöglichen. Unter das vierte Thema fallen alle Erweiterungen, die nicht zu den oben genannten gehören. Dazu gehören zum Beispiel Analog/Digital(A/D)-Wandler beziehungsweise Digital/Analog(D/A-)Wandler.

über alle Grenzen hinaus

Bild 2. Erweiterungsplatine mit drei Steckplätzen für den VC 20. Die Steckplätze sind einzeln zuschaltbar. Man muß darauf achten, daß man nicht gleichzeitig Module benutzt, die den gleichen Adreßbereich belegen.

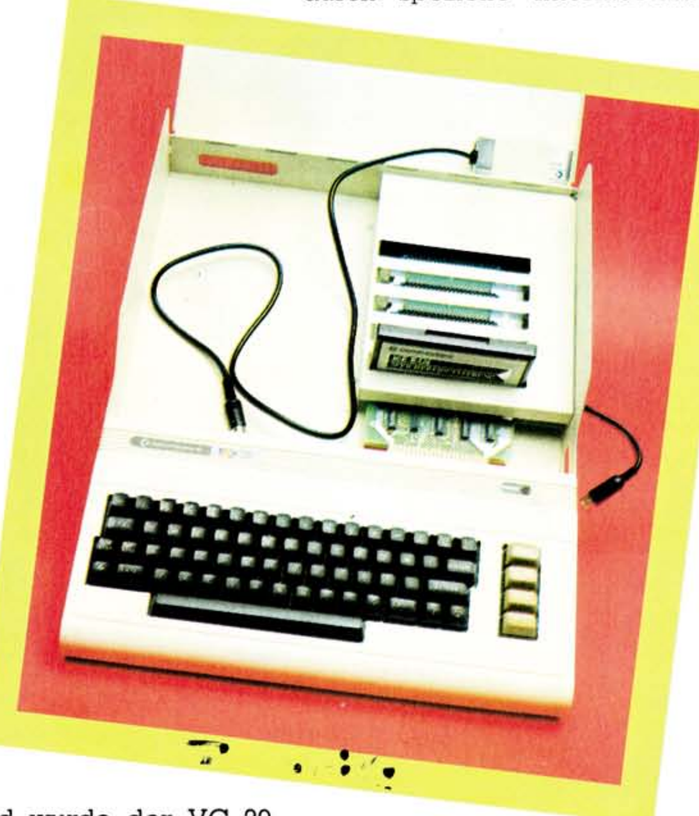


Bild 3. Die Commodore-Modulbox VC 1020. Sie enthält fünf Steckplätze für Module. Der VC 20 wird einfach in die Modulbox integriert.

Hier im Bild wurde der VC 20 herausgezogen. Oben rechts erkennt man den HF-Modulator.

Da der VC 20/C 64 außer seiner seriellen Schnittstelle keine der sonst üblichen Schnittstellen besitzen, ist man gezwungen, wenn man

Schnittstellen/ Interfaces

Geräte, die nicht von Commodore angeboten werden, anschließen will, eine Verbindungsmöglichkeit herzustellen. Das erreicht man durch spezielle Interface-Karten. Man schafft damit eine Verbindung zu:

1. IEC-Schnittstelle, auch IEEE-488-Schnittstelle genannt. Diese Schnittstelle besitzen hauptsächlich alle größeren Commodore-Computer und Peripheriegeräte, aber auch Hewlett-Packard-Geräte (die IEC-Bus-Schnittstelle wurde von Hewlett-Packard entwickelt und heißt dort HP-IB-Hewlett Packard Interface Bus). Damit kann man also auch auf eine große Anzahl von Meßgeräten zugreifen.

IEC-Schnittstellen werden sowohl für den VC 20 als auch für den C 64 angeboten. Mit ihnen kann man also zum Beispiel die größeren Commodore-Diskettenlaufwerke benutzen. Man schließt das Interface einfach an den Expansionsport an. Die unterschiedliche Auslegung dieses Ports beim VC 20 und C 64 erfordert auch eine unterschiedliche Konzeption der Interfacekarten. Ob diese unterschiedliche Auslegung allerdings den um zirka 50 Mark höheren Preis für die Karte des C 64 berechtigt, bleibt dahingestellt. Manche Interfacekarte für den VC 20 besitzt zusätzlich noch einen Sockel für ein 4- oder 8-KByte-EPROM. Das erscheint deshalb sinnvoll, weil beim VC 20 noch freier Adreßraum vorhanden ist.

2. Centronics-Schnittstelle. Gerade für viele Drucker bildet diese Schnittstelle die einzige Verbindung zum Computer.

3. V.24- oder RS232C-Schnittstelle. Auch diese serielle Schnittstelle ist im Mikrocomputerbereich häufig anzutreffen. Sie wurde in Deutschland in der DIN 66020 genormt. Daß man trotz vorhandener Norm statt der Bezeichnung V.24 häufig RS232C liest, könnte daran liegen,

Produktbezeichnung	Preis in DM	geeignet für	Anbieter
Interfaces			
Centronics-Schnittstelle	130,—	VC 20/C64	Bockstaller
IEEE-Schnittstelle	198,—	VC 20/C64	Bockstaller
IEC-Interface	249,—	VC 20	Klaus Jeschke
Recorder-Interface	49,—	VC 20/C64	Klaus Jeschke
Interpod, ein universellesf Interface mit vielen Möglichkeiten	498,—	VC 20/C64	Data Becker
Druckerinterface Centronics parallel	38,50	VC 20/C64	Data Becker
V.24-Schnittstelle	128,—	Commodore-Drucker	Data Becker
Centronics-Eingang für VC 20/C64-Drucker	298,—	VC 20/C64	Data Becker
IEC-Bus-Modul VC 20 = 198 Mark, C64 = 248 Mark	248,—	C64	Data Becker
Interpod, Universal Interface	580,—	VC 20/C64	Schaal Informatic GmbH
Centronics-parallel-Interface im Userport	150,—	VC 20	Brockhaus & Müller GmbH
KFC-Super (Centronics-Schn. + Masch. Monitor + Toolkit + 10mal schnellere Kassettenroutine Kabel für Centronics	198,—	C 64	KFC
	60,—	VC 20/C 64	KFC
Steckkarten für Modulerweiterungen			
Steckplatine 4fach-Modulsteckplatz	175,—	VC 20/C64	Bockstaller
EPROM-Platine für Modulsteckplatz	69,—	VC 20/C64	Bockstaller
I/O-Port-Module	198,—	VC 20/C64	Bockstaller
I/O-Port-Module	495,—	VC 20/C64	Bockstaller
Multiboard-Platine	1875,—	VC 20/C64	Bockstaller
32-KByte-RAM-Modul	179,—	VC 20	Klaus Jeschke
Steckbox für 3 Module	198,—	VC 20	Klaus Jeschke
Steckadapter für 3 Module	99,—	VC 20	Klaus Jeschke
64-KByte-RAM-Modul	338,—	VC 20	Strie
Bus-Platine, 6 Steckplätze, 3 KByte, EPROM-Steckplatz	198,—	VC 20	Strie
Winkeladapter für 2 Module	99,—	VC 20	Data Becker
Modul-Box VC 1020 bis 6 Module	389,—	VC 20	Data Becker

daß die Geburtsstätte der Mikrocomputer in den USA liegt (dort von der Electronic Industrie Association (EIA) als EIA RS232C eingeführt).

Modul und Steckboxen

VC 20/C 64 besitzt bekanntlich nur einen Erweiterungsschacht (Expansion-slot). Da es vorkommen kann, daß man mehrere Module zur gleichen Zeit benutzen möchte (etwa eine Speichererweiterung, eine Spracherweiterung und eine Schnittstelle), wurden Modul- und Steckboxen entwickelt. Sie erlauben die (gleichzeitige) Benutzung mehrerer Steckmodule.

Spätestens dann, wenn man ein professionelles Textverarbeitungsprogramm benutzen will, wünscht man sich, daß der betreffende Computer 80 Zeichen auf den Bildschirm bringt. Erst dann sieht man den Text so, wie er auf dem Drucker erscheinen soll. Dies ermöglichen die 80-Zeichen-Karten.

80-Zeichen-Karte

Leider ist die Benutzung dieser Karten nur in Verbindung mit dem Gebrauch eines Monitors sinnvoll. Ein »normales« Fernsehbild kann aufgrund technischer Gegebenheiten die erforderliche Auflösung nicht bieten. Bevor man sich allerdings diese 80-Zeichen-Karte und den Monitor anschafft, sollte man sich vergewissern, daß auch entsprechende Software verfügbar ist (es sei denn, man programmiert sich diese selbst).

Sonstige Erweiterungen

Hierzu gehören zum Beispiel Analog/Digital-(A-D-) oder Digital/Analog-(D-A-) Wandler. A-D-Wandler bilden eine Schnittstelle zur analogen Umwelt. Das heißt mit diesen Geräten ist man zum Beispiel in der Lage, Meßresultate aufzunehmen und mit dem Computer zu verarbeiten. Es werden die Spannungen der Meßgeräte aufgenommen und in den Wandlern umgesetzt in digitale Werte, die der Computer verstehen kann. D-A-Wandler setzen dementsprechend digitale Werte des Computers um in analoge Spannungen, um dann damit Geräte wie zum Beispiel Motoren zu steuern. (gk)

Produktbezeichnung	Preis in DM	geeignet für	Anbieter
Interfaces			
Modul-Box für 3 Steckplätze	89,—	VC 20	Data Becker
Modul-Box + 8-KByte-RAM	139,—	VC 20	Data Becker
6 Steckplätze	198,—	C64	KFC
Erweiterungsplatine 5 Steckplätze	212,—	C64	Kalawsky
Bausatz	176,—		
64-KByte-RAM-Modul	239,—	VC 20	Roos electronic
Steckplatz für 5 Plätze	139,—	VC 20/C64	Roos electronic
Steckplatz für 2 Plätze	69,—	VC 20/C64	Roos electronic
Busplatine für 6 Module + 3 KByte RAM	198,—	VC 20	KFC
Erweiterungsplatine für 3 Module	125,—	VC 20	KFC
80-Zeichen-Karte			
40/80-Zeichen-Modul	348,—	VC 20	Strie
80-Zeichenkarte-Modul Centronics	248,—	VC 20	Bockstaller
80-Zeichenkarte-Modul Centronics	198,—	C64	Bockstaller
80-Zeichenkarte-Modul	249,—	VC 20	Klaus Jeschke
80-Zeichen-Modul Maxi, VC 20 = 398,—, C64 = 448,— DM	448,—	VC 20/C64	Data Becker
80-Zeichenkarte	279,—	C64	Roos electronic
Sonstige			
A/D-Wandler 8 Bit	120,—	VC 20/C64	Bockstaller
D/A-Wandler 8 Bit	80,—	VC 20/C64	Bockstaller
A/D-Wandler 8 Bit, 16-Kanal	290,—	VC 20/C64	Bockstaller
A/D-Wandler 12 Bit	273,—	VC 20/C64	Bockstaller
A/D-Wandler 12 Bit, Fast Conv.	560,—	VC 20/C64	Bockstaller
Schaltinterface 220 V	185,—	VC 20/C64	Bockstaller
Quickfinger, steuert Joystick am Controlport	49,—	VC 20/C64	KFC

Vorläufige Marktübersicht der verschiedensten Erweiterungen für den VC 20/C 64. Wir wollen diese in den nächsten Ausgaben ergänzen und auch um Software für beide Systeme erweitern. Hersteller, Händler und Leser sind aufgerufen, uns entsprechende Informationen zu liefern.

Tips für sauberes Programmieren

Programmieren ist bekannterweise eine ausgesprochen kreative Tätigkeit, die viele mit Begeisterung ausüben. Erhalten Sie sich diese Freude durch die Anwendung einiger nützlicher Regeln.

Mal im Ernst, haben sie nicht auch schon Programme gesehen oder auch selbst erstellt, die im höchsten Maße unleserlich sind? Vor allem, wenn diese Programme in Basic geschrieben sind, vermißt man des öfteren eine gewisse Übersicht: Da wird ohne ein Konzept wild drauflos getippt, am Anfang weiß man gar nicht so recht, was aus der Programmidee eigentlich mal werden soll. Man fängt ganz locker an, und zuerst klappt alles auch sehr gut. Wenn dann die ersten Erfolge vorhanden sind, denkt man bei sich, daß das Programm ja eigentlich etwas mehr können müßte, oder man bemerkt einige Fehler, die sich während des Programmlaufs einschleichen. Nun beginnt man, kleine Routinen zu entwickeln, die dann an das Ende des Programms angehängt werden, oder, was noch schlimmer ist, irgendwo innerhalb des Programms, wo sie an sich nichts zu suchen haben. Schließlich hat man ja den GOTO-Befehl, der eventuelle Probleme wirksam »umgeht«. Und so entsteht dann mit der Zeit und mit wachsendem Programm eine kaum noch zu übersehende Aneinandereihung von Programmzeilen, gespickt mit GOTO-Befehlen. Wenn solche Programme dann veröffentlicht werden, hat der interessierte Leser zwar die Möglichkeit, dieses Produkt abzutippen, aber wenn er die Programmlogik erkennen und nachvollziehen will, stößt er auf allergrößte Schwierigkeiten. Da hilft manchmal auch eine einigermaßen ausführliche Programmbeschreibung nicht viel. Es soll allerdings Programmierer geben — und das sowohl bei den Amateuren als auch bei den sogenannten Profis — die allerhöchsten Wert darauf legen, von keinem durchschaut zu werden. Außerdem trauen sie sowieso keinem anderen eine Beurteilung ihrer Programme zu. Daß man denen einen schlechten Programmierstil vorwerfen kann, stört sie dann natürlich auch nicht. (Es gibt Fälle, wo Programmierer sich unkündbar gemacht haben, weil kein Mensch außer ihnen selbst das Programm begreift.)

Versuchen Sie dann mal, solch ein Programm zu erweitern, sinnvoll eine zusätzliche Funktion zu implementieren! Auch wenn Sie das Programm selbst »entworfen« haben, und dann nicht peinlich genau Buchführung über jeden Schritt geführt haben, sind Sie nach einem Jahr mit Sicherheit nicht mehr in der Lage, Ihr eigenes Produkt zu verstehen, geschweige es sinnvoll zu ändern beziehungsweise zu erweitern.

Es gibt aber Möglichkeiten, diese Schwierigkeiten zu verringern. Eine Möglichkeit davon ist die strukturierte Programmierung.

Ein Programm ist in der Regel eine Folge von Anweisungen, die der entsprechende Rechner ausführt. Ganz am Anfang der »Computerei« war man beschränkt auf eine sequentielle Methode der Ausführung, Das heißt, jeder Befehl wurde in der Reihenfolge ausgeführt, wie er auch im Programm vorkam. Eine Verzweigung zu einer anderen Stelle oder eine Wiederholfunktion gab es da noch nicht. Das führte dazu, daß diese Programme sehr starr waren. Man konnte keinen direkten Einfluß auf ihren Ablauf nehmen. Programmteile, die mehrmals vorkamen, mußten genauso oft eingegeben werden, wie sie benötigt wurden. Heute kennt jeder die Befehle, die Alternativen zum statischen Programmablauf zulassen. In Basic sind das die Befehle »GOTO« und »GOSUB«.

Programmanweisungen, die den Kontrollfluß bestimmen, zum Beispiel GOTOs, sind also die Ursache dafür, daß die Anweisungen eines Programms in einer anderen als der aufgeschriebenen Reihenfolge ausgeführt werden können (statisch-dynamisch). Ziel der strukturierten Programmierung ist es, durch eine disziplinierte Vorgehensweise die Fehleranfälligkeit zu reduzieren. In anderen höheren Programmiersprachen bedeutet dies zum Beispiel den Verzicht auf GOTOs. An dessen Stelle treten dann einige wenige andere logische Grundstrukturen. In erster Linie handelt es sich um die **Sequenz** von Operationen, **die Auswahl** IF...THEN...ELSE (Verzweigung mit einer oder zwei Bedingungen) und **die Wiederholung** DO..WHILE (einer Gruppe von Operationen, solange eine bestimmte Bedingung erfüllt ist). Neben diesen Grundstrukturen dürfen noch einige weitere Strukturen verwendet werden, im Regelfall jedoch nicht die unbedingte Verzweigung (GOTO).

Der Vorteil: Der Code ist sehr übersichtlich gruppiert und daher auch für andere Programmierer leicht lesbar. Das Testen von strukturiertem Code ist einfach. Strukturierter Code ist leichter wartbar als unstrukturierter.

Der Nachteil: Strukturierte Programme können durch den Verzicht auf GOTO-Anweisungen und durch Codewiederholungen umfangreicher werden als äquivalente, nichtstrukturierte Programme.

Nun besitzt das normale Standard-Basic diese Strukturen nicht. Wer aber als C64-Besitzer in der glücklichen Lage ist, die von Commodore angebotene Basic-Erweiterung Simons Basic sein eigen zu nennen, findet dort einige dieser Befehle (siehe Bericht in dieser Ausgabe). Auch die dort kritisierte Einschränkung des RENUMBER-Befehls wird somit gegenstandslos: Simons Basic erlaubt weitgehend eine vollstrukturierte Programmierung mit dem Verzicht auf GOTOs und GOSUBs. Das bedeutet, daß kein Sprung auf eine Programmzeile xyz mehr nötig ist. Sprungadressen erhalten einen Namen und auch Prozeduren (Unterprogramme) werden mit einem Namen aufgerufen.

Aber unabhängig davon, ob Sie mit oder ohne Simons Basic arbeiten, einige Regeln sollte jeder befolgen.

Grundregel: Der Code soll einfach, klar und übersichtlich (nachvollziehbar) sein. Dazu gehören:

- Die Verwendung einfacher sprachlicher Mittel. Stehen zur Formulierung einer Aktion verschiedene sprachliche Mittel zur Verfügung, sollte das einfache gewählt werden. Das bedeutet: Verzicht auf undurchsichtige Programmierung!
- Das Einrücken von Befehlsfolgen zur Verdeutlichung von Programmzusammenhängen bei geschachtelten Ablaufstrukturen soweit es möglich ist. Anweisungen gleicher Schachtelungstiefe sollten direkt untereinander geschrieben werden (Bild 1.)
- Einfügen von Trennlinien zur optischen Trennung von in sich abgeschlossenen Komponenten. Die sollte man vor allem bei langen Programmen vorsehen. Sie

```
100 REM ********************************
110 REM * PROGRAMM
120 REM * EINGABE,SORTIEREN + AUSGABE *
130 REM ********************************
140 :
145 K=50           :REM ANZAHL DATEN
150 DIM FF$(K+1)
155 :
160 GOSUB 1000    :REM BESETZEN FF$()
165 :
170 OPEN1,4:CMD1  :REM DRUCKER
180 GOSUB 3000    :REM AUSGABE DRUCKER
190 PRINT#1:CLOSE1
195 :
200 GOSUB 2000    :REM SORTIEREN
205 :
210 GOSUB3000:REM AUSGABE BILDSCH.
220 END
230 :
240 :
250 :
260 :
270 :
1000 REM ------------------
1010 REM -           SUBROUTINE       -
1020 REM - BESETZEN FF$() MIT BUCHST. -
1030 REM ----------------------------
1040 :
1050 :
1060 FOR I=1 TO K
1070 :  FF$(I)=CHR$(INT(RND(0)*26)+65)
1080 NEXT I
1090 RETURN
1100 :
1110 :
2000 REM ----------------------------
2010 REM -             SUBROUTINE     -
2020 REM - SORTIEREN VON FF$(1 BIS K) -
2030 REM ----------------------------
2040 :
2050 :
2060 FOR J=1 TO K-1
2070 :  FOR L=J+1 TO K
2080 :     IF FF$(J)<=FF$(L) THEN 2120
2090 :     A$     =FF$(J)
2100 :     FF$(J)=FF$(L)
2110 :     FF$(L)=A$
2120 :  NEXT L
2130 NEXT J
2140 RETURN
2150 :
3000 REM ----------------------------
3010 REM -           SUBROUTINE     -
3020 REM - AUSGABE FF$(1 BIS K)     -
3030 REM ----------------------------
3040 :
3050 :
3060 FOR I=1 TO K
3070 :  PRINT FF$(I);
3080 NEXT I
3090 RETURN
3100 :
```

OK

Bild 1. Der einzige Unterschied zum Bild 2 sind die eingefügten Kommentar-/Leerzeilen und eine andere Aufteilung des Programms. Die Reihenfolge und auch der Algorithmus der ausgeführten Tätigkeiten Besetzen, Sortieren und Ausgabe sind identisch. Aber ist diese Form der Darstellung nicht wesentlich verständlicher und übersichtlicher? Hier Erweiterungen anzufügen oder Teile zu ändern, dürfte keine Schwierigkeiten bereiten. Über diese Art der Aufteilung (in Unterprogramme) berichten wir in einer der nächsten Ausgaben.

erhöhen die Lesbarkeit beträchtlich (Bild 1).

☐ Verwendung von Kommentar als Ergänzung des Codes im Hinblick auf die Problemstellung, nicht als Beschreibung des Codes. Das ist nicht nur bei Assemblerlistings sinnvoll. Der Kommentar zu:

POKE 53281,0 :REM Speicheradresse 53281 mit 0 besetzen

ist sicherlich nicht so sinnvoll wie

POKE 53281,0 :REM Bildschirmfarbe = Schwarz.

☐ Pro Programmzeile nur eine Anweisung. Sie sollten nicht versuchen, möglichst viele Befehle in eine Programmzeile hineinzupressen, wenn Sie sich keine Sorgen um den verfügbaren Speicherplatz machen brauchen.

☐ Die Größe eines Unterprogramms sollte eine Listingseite nicht überschreiten, ausschließlich der Kommentare.

☐ Unterprogramme sollen nur einen Eingang und nur einen Ausgang haben. Und das möglichst am physikalischen Anfang beziehungsweise Ende des Unterprogramms.

☐ Aufgerufene Unterprogramme müssen zum Aufrufpunkt zurückkehren. Verlassen Sie kein Unterprogramm mit GOTO! (Es sei denn zum Abbruch des Programms.)

☐ Benutzen Sie nie eine Variable für mehr als einen Zweck! Wenn in einem Teil des Programms zum Beispiel die Variable AL die Bedeutung Alpha für einen Winkel besitzt, aber im anderen Teil die Bedeutung: Alter hat, verwirrt es doch sehr, und Änderungen sind sehr fehleranfällig!

☐ Ändern Sie nie eine Laufvariable innerhalb der Schleife! (In der Anweisung: FOR I=1 TO 100:PRINT I: NEXT I ist »I« die Laufvariable.)

☐ Sprunganweisungen, die mehr als eine Listingseite auseinanderliegen, tragen sehr zur Unübersichtlichkeit bei.

☐ Vermeiden Sie es, mehr als zwei logische Vergleiche in eine Zeile zu setzen. Es ist sehr schwer, solche Zeilen mit einer Anzahl von logischen Verknüpfungen nachzuvollziehen.

If A AND B OR C AND D AND B<C AND D OR A THEN END

Es bereitet nicht nur sehr viel Mühe, solch eine Programmzeile zu verstehen, Sie werden auch Probleme haben, sie in einem Flußdiagramm sinnvoll darzustellen!

Wenn Sie sich an diese Regeln halten, werden Sie auch nach längerer Zeit noch in der Lage sein, Ihre Programme zu bearbeiten oder sie anderen zu erklären. Und auf diese Art erstellte und veröffentlichte Programme geben auch unseren Lesern eine wertvolle Hilfestellung. (gk)

(Fortsetzung folgt)

```
100 K=50:DIMFF$(51)
110 FORI=1TOK:FF$(I)=CHR$(INT(RND(0)*26)+65):NEXT
120 OPEN1,4:CMD1:GOSUB170
130 PRINT#1:CLOSE1
140 FORJ=1TOK-1:FORL=J+1TOK:IFFF$(J)<=FF$(L)THEN160
150 A$=FF$(J):FF$(J)=FF$(L):FF$(L)=A$
160 NEXTL:NEXTJ:GOTO180
170 FORI=1TOK:PRINTFF$(I);:NEXT:RETURN
180 GOSUB170
```

Bild 2. So sollte es nicht gemacht werden. Dieses Programm ist zwar vom Speicherbedarf her um einiges kürzer als das von Bild 1, die Anzahl der Befehle ist jedoch fast identisch! Aber wissen Sie hier sofort, um was es geht?

Simons

Eine notwendige Erweiterung für den Commodore 64

Erfahrene Commodore-Programmierer werden mir sicherlich zustimmen: Die ausgezeichneten Commodore-Editiermöglichkeiten verhalten sich für den Programmierer umgekehrt proportional zu den Basic-Versionen. Dies ist auch bei dem Commodore 64 nicht anders. Obwohl der C 64 über eine hochauflösende Grafik verfügt, bietet das Standard-Basic hier keine Unterstützung. Für häufige Programmierarbeiten sind Basic-Erweiterungen — insbesondere für den Grafikteil und die Sprites — eine notwendige Hilfe. Eine solche Erweiterung ist Simons Basic für den Commodore 64.

Teil 1



Simons Basic bietet sehr viele wichtige Befehle. Bild 1 zeigt eine Übersicht über alle Befehle und eine Kurzbeschreibung ihrer Bedeutung. Diese Übersicht kann auch als Handzettel für diejenigen dienen, die schon mit Simons Basic arbeiten.

Simons Basic enthält viele dringend notwendige Befehle, aber auch Befehle, die wohl nur in sehr seltenen Fällen benutzt werden. Auf jeden Fall ist Simons Basic für den geübten Programmierer eine wertvolle Unterstützung. Besonders hervorzuheben sind hier die Befehle, die in Bild 1 unter Programmierhilfen zusammengestellt sind, die in dieser oder ähnlicher Form auch schon von anderen Programmier-Kits her bekannt sein dürften. Weiterhin einige Befehle zur Verarbeitung von Zeichenreihen wie zum Beispiel INST. Für Programmierer, die auch andere Programmiersprachen wie PL/1 oder Pascal kennen, dürften besonders die neuen Strukturbefehle und die ERROR-Befehle interessant sein.

Um die speziellen Möglichkeiten des Commodore 64 wie die hochauflösende Grafik, die Definition von Sprites und den Sound-Generator zu benutzen, sind natürlich die entsprechenden Befehle notwendige Voraussetzung, wenn Programmieren nicht in Byte-Fummelei ausarten soll.

Zu den Befehlen, die wohl nur dann angewendet werden, wenn ein Programmierer auch alle Register des Computer sehen will, gehören neben einigen Befehlen aus den anderen Bereichen bestimmt auch alle Befehle der Bildschirmsteuerung.

Alles in allem kann man jedoch sagen: Der zusätzliche Befehlsvorrat von Simons Basic läßt fast keine Wünsche offen.

Gehen wir im folgenden kurz auf die verschiedenen Befehle und ihre Anwendungsmöglichkeiten ein:

Programmierhilfen

AUTO
Dieser Befehl ist von anderen Kits bestimmt schon hinlänglich bekannt. Er ermöglicht die zeilenweise Programmeditierung, ohne jeweils eine neue Zeilennummer miteintippen zu müssen. Dies erspart hauptsächlich beim fließenden Eintippen eines Programms die Überlegung: Welches ist denn jetzt die nächste Zeile?

COLD
Dieser Befehl ersetzt das Ein- und Ausschalten des Computers, wenn ein Kaltstart durchgeführt werden soll. Intern werden im Computer immer Zeiger verwaltet, die auf den Anfang des Programms, den Anfang der Variablenbereiche und so weiter zeigen. Der Befehl COLD bewirkt nichts anderes als das Rücksetzen dieser Zeiger in den Ausgangszustand.

DELAY
Mit dem Befehl DELAY kann die Listgeschwindigkeit eingestellt werden. Daß hier 256 Möglichkeiten zur Verfügung stehen, ist mehr als ein Programmierer benötigt. Prinzipiell wird sich jeder aus den Möglichkeiten ein oder zwei Geschwindigkeiten aussuchen, die seiner Lesegeschwindigkeit am Bildschirm entsprechen.

DISAPA
In Verbindung mit dem Befehl SECURE ist der Befehl DISAPA ein hinreichend wirkungsvolles Mittel, um sein Programm gegen unbefugtes Auflisten zu schützen. Im Prinzip wäre es möglich, das gesamte Programm mit diesem Befehl zu schützen, jedoch macht man sich selbst die Arbeit der Softwarepflege damit nur schwieriger. Sinnvoll wäre es, diesen Befehl in einem kurzen Programmstück zu verwenden, welches einige andere Sicherungsmethoden enthält.

DISPLAY
Eine reine Informationsanweisung, die aber sehr wichtig ist, da es sonst sehr schwierig wäre, die aktuelle Belegung der Funktionstasten festzustellen.

DUMP
Der Vorteil eines Interpreters liegt zu einem großen Teil darin, daß während eines Programmlaufes das Programm abgebrochen werden kann und die Variablen abgefragt werden können. Dies erleichtert das Austesten erheblich gegenüber Compilerversionen. Nun ist es recht mühsam, immer nach einem BREAK im Programm einen PRINT-Befehl für alle — oder auch nur die benötigten — Variablen einzugeben, wenn mehrere sogenannte BREAK-POINTS gesetzt sind. Diese Arbeit erleichtert der DUMP-Befehl.

FIND
Ähnlich dem DUMP-Befehl erleichtert der FIND-Befehl das Testen sowie das Dokumentieren von Programmen. Besonders bei langen Listings ist es sehr mühsam, das gesamte Programm nach einer bestimmten Variablen zu durchsuchen. Da in Basic auch im Prinzip alle Variablen global sind, dürften — außer temporären Variablen — den Variablen nicht mehrfache Bedeutungen zugewiesen werden. Mit dem FIND-Befehl ist es unter anderem möglich zu prüfen, ob eine Variable schon im Programm vorhanden ist oder nicht.

KEY
Da der Commodore 64 Funktionstasten anbietet, ist es auch sinnvoll, diese mit häufig verwendeten Basic-Befehlen (zum Beispiel LIST) zu belegen.

Simons Basic

MERGE

Der MERGE-Befehl ermöglicht zwar das Einkopieren von anderen Programmen in ein Programm, das sich im Hauptspeicher befindet, jedoch läßt dieser Befehl einige Möglichkeiten vermissen. Zum Beispiel ist das Laden von bestimmten Programmteilen eines Programms von Diskette nicht möglich. Dies ist besonders dann ein Nachteil, wenn aus anderen Programmen nur bestimmte Unterprogramme übernommen werden sollen.

OLD

Ab und zu kann es vorkommen, daß versehentlich ein NEW-Befehl eingegeben wurde, und man feststellt, daß das Programm vorher nicht abgespeichert war beziehungsweise die Kontrollampe an dem Floppy Disk-Laufwerk blinkt. Da durch den NEW-Befehl nur Zeiger intern umbesetzt werden, ist eigentlich noch nicht alles verloren. Aber es ist doch sehr mühsam, das Programmende des Programms und die Werte für den Beginn der Variablentabelle und so weiter ausfindig zu machen. Dies erspart einem der OLD-Befehl.

OPTION

Eine Anwendungsmöglichkeit für diesen Befehl, der alle Befehle von Simons-Basic hervorhebt, ist direkt nicht ersichtlich. Nützlich ist er vielleicht, wenn ein Programm in normales Basic umgeschrieben werden soll. Aber wenn jemand ein Programm, das mit Simons Basic erstellt wurde, erhält, und dies umschreiben will, weil ihm die Programmierunterstützung nicht zur Verfügung steht, der könnte diesen Befehl gebrauchen. Aber der hat ja kein Simons Basic. Und wer gibt schon seine Programme weiter mit einer Liste: Hier sind die Befehle, die geändert werden müssen?

PAGE

Da der Bildschirm des Commodore 64 nur 40 Zeichen je Zeile hat und das Auslisten der Programme sehr schnell geht, verschwinden Programmstücke nach oben aus dem Bildschirm heraus schneller, als man eventuell die STOP-Taste gefunden hat. Dies kann man einerseits mit der Benutzung der CTRL-Taste beeinflussen, andererseits mit dem weiter vorn beschriebenen DELAY-Befehl. Komfortabel ist es natürlich, wenn man vor Beginn

Befehlsübersicht Simons Basic:

Programmierhilfen:

AUTO	—	Zeilennummernvergabe bei Programmeditierung
COLD	—	Kaltstart, ersetzt aus-/einschalten
DELAY	—	Listgeschwindigkeit einstellen
DISAPA	—	Anweisung schützen
DISPLAY	—	Belegung der Funktionstasten anzeigen
DUMP	—	Variablen mit Werten anzeigen
FIND	—	Basic-Befehle oder Zeichenreihen im Programm suchen
KEY	—	Funktionstasten mit Basicbefehl belegen
MERGE	—	anderes Programm in bestehendes einkopieren
OLD	—	NEW-Befehl aufheben
OPTION	—	Simons Basicbefehle hervorheben
PAGE	—	seitenweise Listenausgabe
RENUMBER	—	Zeilen umnumerieren (ohne Zeilenangaben bei GOTO und GOSUB)
SECURE	—	Programmzeile schützen
TRACE/RETRACE	—	aktuelle Zeilennummer, die im Programm durchlaufen wird, anzeigen und wieder aufheben

Struktur-Befehle und ERROR-Befehle:

CALL	—	Sprung zu einer mit PROC definierten Routine (ähnlich GOTO)
END PROC	—	Ende einer Routine, ähnlich RETURN
EXEC	—	Unterprogrammaufruf für Routinen die mit PROC und END PROC definiert wurden
GLOBAL	—	ursprünglichen Variablenwert wieder zuweisen
IF..THEN..ELSE	—	Bedingte Anweisung mit doppelter Anwendungsmöglichkeit
LOCAL	—	Block bedingte Variablen
LOOP...EXIT IF ...END LOOP	—	Schleifendurchlauf mit bedingtem Abbruch
NO ERROR	—	Fehlermeldung unterdrücken
ON ERROR	—	Sprungverteile für Fehlermeldungen
PROC	—	Sprungadresse (symbolisch)
RCOMP...ELSE	—	Bedingte Anweisung, wobei die Bedingung von der letzten IF-Abfrage übernommen wird
REPEAT...UNTIL	—	ähnlich FOR...NEXT für bedingte Schleifen

Grafik-Befehle

ANGL	—	Radius zeichnen
ARC	—	Segment zeichnen
BLOCK	—	farbig ausgefülltes Rechteck ausgeben
CHAR	—	Zeichen in Grafik-Bildschirm
CIRCLE	—	Ellipse (Sonderfall: Kreis) ausgeben
CSET	—	Zeichensatz umschalten
DRAW	—	Figur zeichnen
HICOL	—	Nach LOW COL zum zurücksetzen auf die drei Farben, die mit MULTI definiert werden
HIRES	—	hochauflösende Grafik (mit Wahl der Vordergrund- und Hintergrundfarbe) einschalten
LINE	—	Linie zeichnen
LOW COL	—	drei weitere Farben zum Multi-Color-Modus zuschalten
MULTI	—	Multi-Color-Modus mit drei Zeichenfarben bestimmen
PAINT	—	Fläche mit Farbe füllen
PLOT	—	Punkt ausgeben
REC	—	Rechteck zeichnen
ROT	—	Figur drehen
TEST	—	Punkt vorhanden?
TEXT	—	Text in Grafik-Bildschirm

Sprite-Befehle:

(KLAMMERAFFE)	—	Form eines Sprites definieren
CHECK	—	Kollision abfragen
CMOB	—	Farben für Multi-Color-Sprite festlegen
DESIGN	—	Speicherzuteilung für Sprite
DETECT	—	Kollision vorbereiten
MMOB	—	Sprite darstellen oder bewegen
MOB OFF	—	Sprite ausschalten
MOB SET	—	Eigenschaften eines Sprite festlegen
RLOCMOB	—	Sprite bewegen

einer jeden Programmiersitzung den Befehl Page verwendet, womit ein seitenweises Blättern in Vorwärtsrichtung erzielt werden kann.

RENUMBER

Wo fast jedes auf dem Commodore 64 erstelltes Programm dynamisch wächst, wird mal hier eine Zeile eingefügt, mal wird dort eine Zeile herausgenommen. Um dieses ganze Zeilennummern-Wirrwarr in den Griff zu bekommen ist natürlich der RENUMBER-Befehl sehr nützlich. Leider wirkt sich der RENUMBER-Befehl nicht auf solche Zeilennummern aus, die hinter GOTO und GOSUB stehen. In mühsamer Kleinarbeit artet es dann aus, wenn Sie anschließend alle Sprungadressen bei GOTO/GOSUB-Befehl von Hand ändern müssen.

SECURE

Dieser Befehl bewirkt nur das eigentliche Schützen, der durch den Befehl DISAPA gekennzeichneten Befehle.

TRACE/RETRACE

Zum Testen von Programmen — besonders bei sogenannten Endlos-Schleifen — leistet der TRACE-Befehl, mit dem die aktuelle Zeilennummer eines laufenden Programmes angezeigt wird, sehr nützliche Hilfe.

Strukturbefehle und ERROR-Befehle

Diese Befehle lassen sich als Einzelbefehle nicht ausreichend erklären, da sie eine gewisse Blockstruktur voraussetzen, so daß wir diese im Zusammenhang besprechen wollen.

Schleifen/bedingte Schleifen/ bedingte Anweisungen

Der erste Bereich der Strukturbefehle widmet sich den Schleifen und bedingten Anweisungen. Da das normale Basic nur IF...THEN-Befehle zuläßt, ist es eine wesentliche Vereinfachung, wenn diese Befehle auch einen ELSE-Teil erhalten. Dadurch können aufwendige Konstruktionen mit GOTO-Befehl vermieden werden, wie Bild 2 zeigt. Bild 3 zeigt die Bedingungen bei einem IF-Statement, die sehr komplex sein können, so daß es sinnvoll ist, diese Bedingung in einem weiteren Befehl ohne erneute Eingabe wieder prüfen zu können. Dies kann mit dem Befehl

Bild 1. Diese Befehle bietet Simons Basic

Musik-Befehle:

ENVELOPE	—	Hüllkurve einstellen
MUSIC	—	Noten festlegen
PLAY	—	Musikwiedergabe
VOL	—	Lautstärke einstellen
WAVE	—	Wellenform einstellen

Befehle für Zeichenreihen

AT	—	Zeichenreihe auf Bildschirm positionieren
CENTRE	—	Ausgabe einer Zeichenreihe in der Mitte einer Bildschirmzeile
CHAR	—	Zeichen in Grafik-Bildschirm
DUP	—	Zeichenreihe vervielfachen
INSERT	—	Zeichenreihe in andere einfügen
INST	—	Zeichenreihe mit einer anderen überschreiben
PLACE	—	Zeichenreihe in Zeichenreihe suchen
TEXT	—	Text in Grafik-Bildschirm

Befehl für Zahlen:

$	—	Umwandlung Hexadezimal in Dezimal
%	—	Umwandlung Binär in Dezimal
DIV	—	Division ohne Rest
EXOR	—	bitweise Verknüpfung von Zahlen mit EXKLUSIV ODER
FRAC	—	Nachkommastellen einer Dezimalzahl

Bildschirmsteuerung

BFLASH	—	Farbwechsel Bildschirmrahmen einschalten
BFLASH O	—	Farbwechsel Bildschirmrahmen ausschalten
COPY	—	Hardcopy einer hochauflösenden Grafik
DOWN	—	Bildschirmbereich nach unten rollen
FCHR	—	Bildschirmbereich mit Zeichen füllen
FCOL	—	Zeichenfarbe in Bildschirmbereich bestimmen
FLASH	—	Blinken einer Bildschirmfarbe einschalten
FILL	—	Bildschirmbereich mit Farbe und Zeichen füllen
HRDCPY	—	Hardcopy eines normalen Bildschirmes
INV	—	Bildschirmbereich invertieren
LEFT	—	Bildschirmbereich nach links rollen
MOVE	—	Bildschirmbereich duplizieren
OFF	—	Blinken einer Bildschirmfarbe ausschalten
RIGHT	—	Bildschirmbereich nach rechts rollen
SCRLD	—	Bildschirm (der mit SCRSV gespeichert wurde) laden
SCRSV	—	Bildschirm (Normal-Modus) speichern
UP	—	Bildschirmbereich nach oben rollen

Befehle für Light-Pen, Joystick und Paddle

JOY	—	Funktion des Joystick bestimmen
PENX	—	X-Koordinate des Light-Pen
PENY	—	Y-Koordinate des Light-Pen
POT	—	Widerstand Paddle feststellen (Potentiometer)

Sonstige Befehle

(KLAMMERAFFE)	—	neues Zeichen definieren
DESIGN	—	neu zu erstellendes Zeichen festlegen
DIR	—	Inhaltsverzeichnis einer Diskette ganz oder teilweise (Jokerzeichen) anzeigen
DISC	—	Diskbefehl ausführen
FETCH	—	Kontrollierte Eingabe
INKEY	—	Abfrage auf gedrückte Funktionstaste
LIN	—	aktuelle Zeile des Cursors anzeigen
MEM	—	Zeichensatz von ROM-Bereich in RAM-Bereich verlegen
PAUSE	—	Pause im Programm (ersetzt »leere« FOR...NEXT—Schleife)
RESET	—	Zeiger auf beliebige DATA-Zeile setzen

Simons Basic

RCOMP...ELSE, da nicht wie in anderen Programmiersprachen eine blockweise Bearbeitung in verschiedenen Zeilen der THEN-/ELSE-Teile erfolgen kann, substituiert werden.

Eine weitere Verbesserung ist die Programmierung von Schleifen mit Bedingungsteilen. Das Beispiel im Handbuch ist relativ ungünstig gewählt, da dieses Beispiel durch eine einfache FOR...NEXT-Schleife ersetzt werden kann. Bild 4 zeigt einen sinnvollen Einsatz für den Befehl REPEAT...UNTIL. Dabei wird die Schleife abgebrochen, wenn eine Bedingung erfüllt ist, die nicht in einer FOR...NEXT-Schleife einprogrammiert werden kann. Sicherlich ist es auch bei einfachen FOR....NEXT-Schleifen möglich, diese Schleifen mit einer IF-Abfrage zu verlassen, jedoch wird das Programm durch die neuen Befehle viel übersichtlicher. Ähnliches leistet auch der Befehl LOOP...EXIT IF...END LOOP.

```
100 REM OHNE IF...THEN...ELSE
110 IF A=B THEN C=D : GOTO 130
120 E=F
130 REM FORTSETZUNG
140 :
150 :
160 :
170 REM MIT IF...THEN...ELSE
180 IF A=B THEN C=D ELSE E=F
190 REM FORTSETZUNG
```

Bild 2. Ein Beispiel für IF...THEN...ELSE

```
 90 REM OHNE RCOMP...ELSE
 100 IF A=B AND X=Y OR F<G AND H>J AND NOT Y=R THEN PRINT "SEHR LANGER TEXT";
 110 IF A=B AND X=Y OR F<G AND H>J AND NOT Y=R THEN PRINT "DER NICHT IN EINE";
 120 IF A=B AND X=Y OR F<G AND H>J AND NOT Y=R THEN PRINT "ZEILE PASST. !!!!";
 130 IF A=B AND X=Y OR F<G AND H>J AND NOT Y=R THEN GOTO 150
 140 PRINT"NOCH EIN TEXT"
 150 REM FORTSETZUNG
 160 :
 170 :
 180 :
 190 REM MIT RCOMP...ELSE
 200 IF A=B AND X=Y OR F<G AND H>J AND NOT Y=R THEN PRINT "SEHR LANGER TEXT";
 210 RCOMP PRINT"DER NICHT IN EINE ZEILE PASST !!!!!" ELSE PRINT"NOCH EIN TEXT
 220 REM FORTSETZUNG
READY.
```

Bild 3. Beispiel für RCOMP...ELSE

```
100 REM VERGLEICH ZWEIER ZAHLEN ALS ABBRUCHKRITERIUM
110 REPEAT
120 ZN = ZA / 3
130 UNTIL ABS(ZN-ZA) <0.0000001
```

Bild 4. So wird REPEAT...UNTIL eingesetzt.

Prozeduren

Sehr schön handhaben läßt sich die Verwendung von Unterprogrammen als Prozeduren mit Simons Basic. Wie in blockorientierten Sprachen existiert auch ein Befehl PROC, der praktisch die Marke eines Unterprogrammes ist. Das Unterprogramm (in diesem Fall spricht man von Prozedur) wird auch nicht mit RETURN beendet, sondern mit END PROC. Der Aufruf kann mit CALL oder mit EXEC erfolgen, wobei CALL einem GOTO entspricht (eine unübliche Art des Aufrufs einer Prozedur, da Prozeduren normal unabhängig von ihrer Lage im Programm ausgeführt werden) und EXEC einem GOSUB.

Wenn auch keine Blockvariablen im ursprünglichen Sinne zugelassen sind, kann man jedoch mit dem Befehl LOCAL Variableninhalte retten und später mit dem Befehl GLOBAL wieder auf diese Werte zurückgreifen. Dies erleichtert insbesondere die Programmierung großer komplexer Programme mit vielen Prozeduren.

Fehlerbehandlung

Die beiden Befehle ON ERROR und NO ERROR erlauben eine relativ komfortable Fehlerbehandlung. Die normale Fehlerbehandlung (Programmabbruch mit Anzeige des Fehlers) ist in den meisten Fällen nicht sehr benutzerfreundlich, da die Fehler per Programm abgefangen und durch eine entsprechende Benutzermitteilung eventuell auch behoben werden könnten. Mit dem Befehl ON ERROR ist eine solche komfortable Fehlerbehandlung in Abhängigkeit des aufgetretenen Fehlers (Liste im Handbuch enthalten) möglich. Lediglich eine Unterdrückung der Fehlermeldungen ist durch den Befehl NO ERROR möglich.

In der nächsten Ausgabe werden wir uns mit den Grafik-, Sprite- und Musik-Befehlen von Simons Basic sowie mit den Befehlen für Zeichenreihen, Zahlen, Light-Pen, Joystick, Paddle und der Bildschirmsteuerung beschäftigen.

(H.L. Schneider)

Angreifer aus dem Weltall

Retten Sie New York vor den Monstern aus dem Weltall. Was Sie dazu brauchen? Schnelligkeit und Geschicklichkeit. Oder setzen Sie sich mit einer Horde von Angreifern im Weltall auseinander!

Save New York ist ein neues Spiel-Steckmodul für den Commodore 64 von Creative Software (125 Mark), »Save New York« oder »Retten Sie New York«, bringt ein uraltes Videospielthema in neuer Verpackung. Die Stadt New York soll vor grauslichen Monstern aus dem fernen Weltall beschützt werden. Sobald diese Monster über New York herfallen, fangen sie an, die Wolkenkratzer anzuknabbern. Darüber hinaus legen die Monster zuweilen Riesenneier, die zu Boden fallen und sich in ein Monsterbaby verwandeln, das den gesamten New Yorker Untergrund unsicher macht und die Wolkenkratzer vom Keller herauf verspeist. Das ist natürlich noch viel gefährlicher als die Schäden, die die Luftmonster anrichten können, denn bei einem Untergrundbefall stürzen die Hochhäuser mit Leichtigkeit ein.

Die Aufgabe des Spielers oder der zwei Spieler ist es, alle Luft- und Untergrundmonster loszuwerden, bevor sie ganz New York aufgefressen haben.

Einziges Mittel zur Lösung dieser Aufgabe ist ein, mit dem Joystick in alle acht Himmelsrichtungen steuerbarer kleiner Düsenjäger, mit dem man die Luftmonster abschießen kann. Dabei muß man allerdings aufpassen, denn sobald man von einem der Monster erwischt wird, oder aber gegen einen der Wolkenkratzer fliegt, explodiert das Raumschiff. Auch muß der Spieler darauf achten, immer genügend Treibstoff in seinem Benzinkanister zu haben, sonst stürzt sein kleines Raumschiff unfreiwillig ab. Wenn man die von Tankflugzeugen abgeworfenen »Treibstoff-Fallschirme mit Hilfe seines Düsenjägers auffängt, kann man während des Flugs, das ist übrigens die einzige Möglichkeit, Treibstoff tanken.

Zur Bekämpfung der Untergrund-Monster muß man landen, seinen Piloten in den Untergrund schicken und versuchen, die Monster von dort aus zu erwischen. Allerdings kann man dabei auch von einer der berüchtigten New Yorker U-Bahnen überrollt werden.

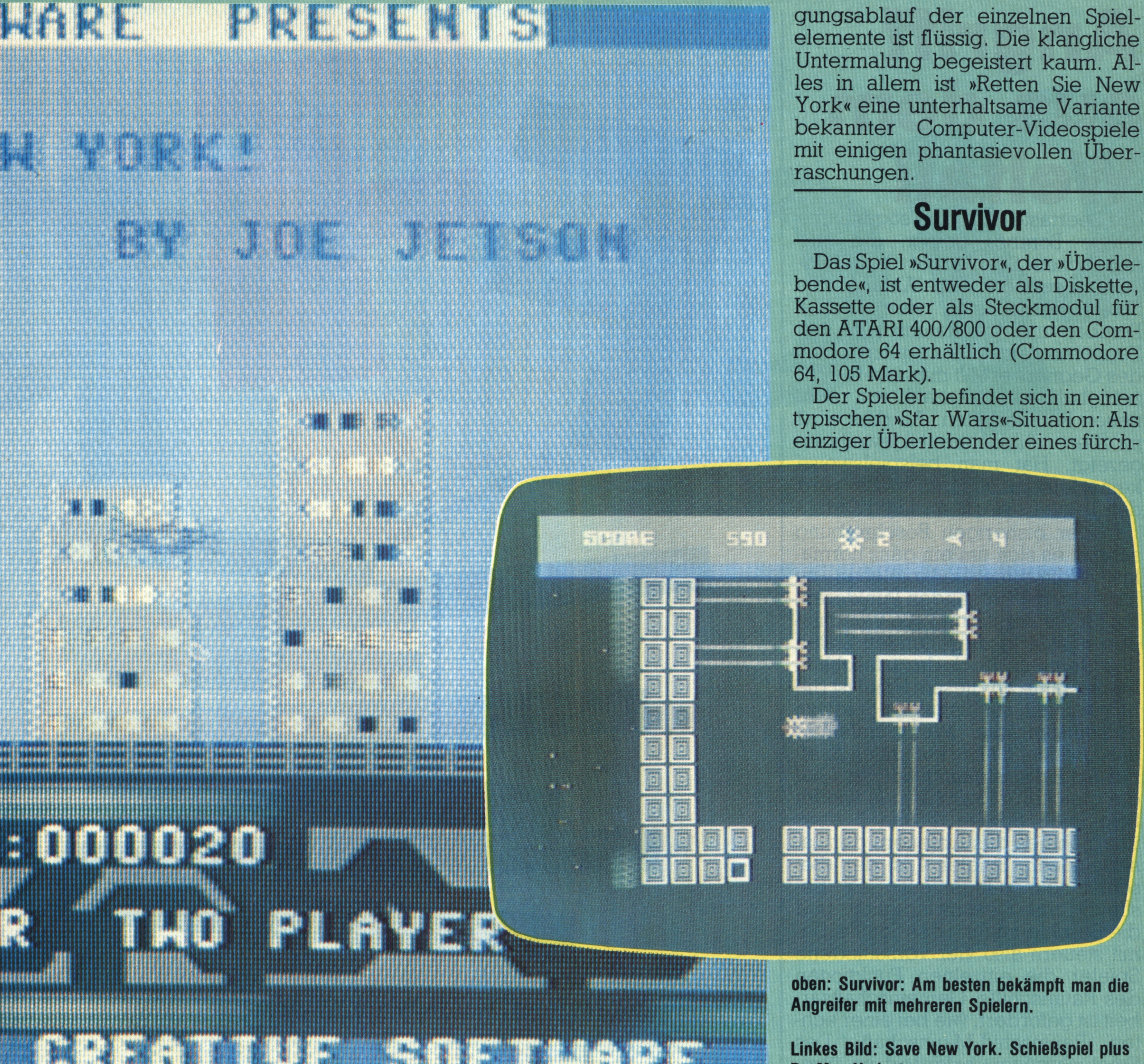
Das Spiel ist nicht einfach, denn die Luftmonster verstecken sich sehr gerne hinter oder zwischen den Wolkenkratzern, wo man sie kaum erwischen kann, ohne höchstpersönlich ganze Stockwerke der

gungsablauf der einzelnen Spielelemente ist flüssig. Die klangliche Untermalung begeistert kaum. Alles in allem ist »Retten Sie New York« eine unterhaltsame Variante bekannter Computer-Videospiele mit einigen phantasievollen Überraschungen.

Survivor

Das Spiel »Survivor«, der »Überlebende«, ist entweder als Diskette, Kassette oder als Steckmodul für den ATARI 400/800 oder den Commodore 64 erhältlich (Commodore 64, 105 Mark).

Der Spieler befindet sich in einer typischen »Star Wars«-Situation: Als einziger Überlebender eines fürch-



oben: Survivor: Am besten bekämpft man die Angreifer mit mehreren Spielern.

Linkes Bild: Save New York. Schießspiel plus PacMan-Variante.

Hochhäuser abzurasieren. Im Untergrund hat man ganz ähnliche Schwierigkeiten, wie viele sie von Pac-Man her kennen: Den Untergrundbahnen auszuweichen, ist nicht immer ganz einfach.

Das Spiel »Save New York« hat mehrere Schwierigkeitsstufen. Schafft man es, die erste Welle der Angreifer abzuwehren, das heißt, 10 Mutanten-Monster erwischt, so hat man die erste Spielrunde überstanden, in jeder weiteren Spielrunde kommen jeweils 16 weitere Monster hinzu. Punkte gibt es je nach Schwierigkeit: 20 für das Abschießen eines fliegenden Monsters, 50 für das Treffen eines Eis, bevor es sich in ein Babymonster verwandelt und 90 für das Erwischen eines Babymonsters. Für Spannung ist also gesorgt.

Das Interessanteste an dem Spiel erscheint mir die Integration zweier verschiedener Spielstrategien in einem. Auf der einen Seite ist »Save New York« ein herkömmliches, zugegeben etwas komplizierteres Schießspiel, indem es um die Rettung einer Großstadt vor feindlichen Weltraummonstern geht. Auf der anderen Seite aber ist es ein phantasiereiches Quasi-Pac-Man Spiel, bei dem es nicht so sehr ums Abschießen, als vielmehr um Geschicklichkeit im Ausweichen vor unvorhergesehenen Gefahren geht.

Grafisch ist das Spiel recht ansehnlich realisiert. Der Bewe-

terlichen Angriffs der Raumflotte der Xenogryphen sieht man sich vier schwerbewachten und stark verteidigten Raumstützpunkten der Xenogryphen gegenüber, die es zu zerstören gilt. Daran wird man durch ganze Schwärme von Kampfschiffen, Flugbomben und ferngesteuerten Asteroiden gehindert, gegen die man sich natürlich auch zur Wehr setzen muß.

Zum Spielstart gibt es einige Einsteigehilfen: mit der Taste A werden die Laserkanonen des eigenen Raumschiffs automatisch abgefeuert (sonst von Hand mit dem Joystickcontroller). Mit der Taste T beschleunigt das Schiff sofort, andererseits hätte es eine natürliche physikalische Beschleunigung. Mit

Angriff aus dem Weltall

der Leertaste wird eine sogenannte »kluge Bombe« gelegt, die alle feindlichen Schiffe auf dem Bildschirm zerstört. Am Anfang des Spiels besitzt man vier solcher Bomben und drei Raumschiffe, bei Zerstörung eines Raumstützpunktes des Gegners erhält man zwei hinzu, zuzüglich einer Anzahl von zwei neuen Raumschiffen. All dies wird im oberen Bildschirmteil jeweils angezeigt. Hat man beispielsweise keine klugen Bomben mehr, wird das Textfenster rot anstatt grün.

In der bisherigen Beschreibung scheint es sich um ein ganz normales Weltraumschießspielchen zu handeln, mit steigenden Punktezahlen für die Zerstörung von verschieden schwierig zu treffenden Zielen des Gegners. Doch das tatsächlich Interessante an diesem Spiel ist die Möglichkeit, mit bis zu vier Spielern gemeinsam die Weltraumfeinde anzugreifen. Dabei kommt es zu einer Art Arbeitsteilung. Spieler 1 hat die gleichen Möglichkeiten wie im Solo-Spiel, Spieler 2 und 3 aber übernehmen das Abfeuern der Laserkanonen. Spieler 4 ist der »Beschleunigungs-Ingenieur«. Er bestimmt die Geschwindigkeit und die Beschleunigung des Schiffs. Damit steuern also vier verschiedene Spieler die einzelnen Funktionen des Raumschiffs, Gemeinschaftsarbeit ist gefordert, wie bei einer echten Raumschiff-Mannschaft. Das Spiel »Survivor« erfordert im übrigen einiges an Geschicklichkeit, denn um die Laserkanonenstellungen des Gegners ins Visier zu bekommen, genügt es nicht, nur den Schutzwall wegzuballern, sondern man muß sehr nahe an die Stellungen heranfahren und zwischen allerlei Beschuß, Wänden, und sonstigen Gefahren herummanövrieren.

Die Grafik des Spiels ist für Heimcomputer außerordentlich gut, zumindest für zweidimensionale Spiele. Klangliche Einführung bildet eine ziemlich verkorkste Version von Wagners »Ritt der Walküren«, ansonsten haben wir das übliche Videospiel-Klingklang. Im großen und ganzen ist »Survivor« ein recht unterhaltsames Weltraumspiel.

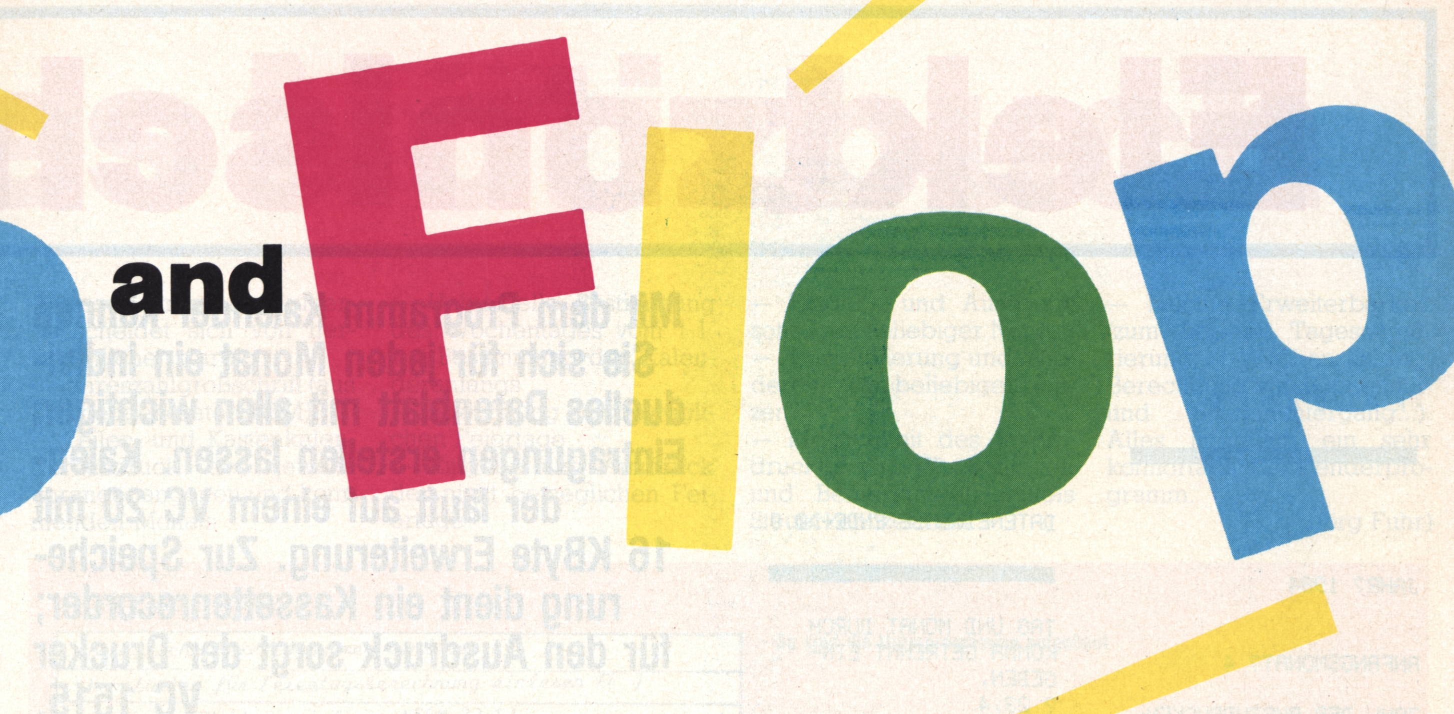
(Stephan Kaske)



Mit »Flip and Flop« wurde ein Spiel auf den Markt gebracht, das auf der Idee von »Q-Bert« basiert. Es wurden jedoch gegenüber dem Vorbild ganz erhebliche inhaltliche und technische Verbesserungen vorgenommen.

Nun zum Spiel selbst: Zwei Figuren, nämlich Flip, das Känguruh, und Mitch, der Affe, müssen abwechselnd über ein System von durch Leitern verbundenen Plattformen dirigiert werden. Selbstverständlich können sie ihre Aufgabe, die markierten Teilstücke der Plattformen zu berühren und dadurch neu zu färben, nicht ungestört ausüben. Flip, das Känguruh, das sich auf den Plattformen hüpfend fortbewegt, wird von einem Zoowärter verfolgt, der es einzufangen versucht.

Mitch, der Affe, der an der Unterseite der Plattformen hängt, muß vor einer Art Gitter fliehen. Glücklicherweise gibt es noch blinkende Flächen, die jeden, der sie berührt, für ein paar Sekunden festhalten. Doch aufgepaßt: Wenn man solch eine Stelle aus Versehen betritt, dann ist man so gut wie erledigt! Hat man fünf Runden überstanden, darf Flip an einer Leiter zu seinen Freunden im Zirkus heruntersteigen, von denen er mit einem



Spruchband (»Welcome back«) begrüßt wird.

Dem Spieler werden verschiedene Spielvarianten angeboten: Einerseits kann zu zweit gespielt werden (mit einem oder zwei Joysticks) und andererseits kann bei einer fortgeschritteneren Spielstufe begonnen werden.

Wie auch bei »Juice!« ist es nicht notwendig, den Joystick diagonal zu bewegen, da beispielsweise ein Hebeldruck nach oben als Sprung nach rechts oben interpretiert wird. Hat man sich erst einmal daran gewöhnt, dann kommt man mit dieser Steuerung sehr gut zurecht.

Wenn in späteren Runden das Spielfeld so groß wird, daß es nicht mehr auf den Bildschirm paßt, macht es sich positiv bemerkbar, daß das Spielfeld entsprechend der Bewegung des Spielers auf dem Bildschirm verschoben wird.

Auch sonst ist die grafische Darstellung aufgrund der dezenten Farben und der detaillierten und farbigen Darstellung der bewegten Objekte ausgezeichnet.

Die akustische Untermalung ist sehr gut gelungen, wenn man mal davon absieht, daß vor Beginn jeder Runde immer wieder dieselbe eintönige Melodie gespielt wird, was auf die Dauer sehr ermüdet.

Dennoch muß man sagen, daß »Flip and Flop« von First Star Software ganz klar im Vergleich zu »Juice!«, »Slinkey« und »Q-Bert« der beste Vertreter dieser Spielgattung auf Commodore 64 (es gibt auch eine Atari-Version) ist. Der Preis liegt bei 140 Mark.

(Julian Reschke)

Elektronisch

```
K A L E N D E R

JAHR? 1984

ANFANGSMONAT? 4

ZAHL DER AUSZUDRUCKEN-
DEN MONATE
? 2

DATENEINGABE

1 - DATENEINGABE

2 - ZUSAETZLICH SPEI-
    CHERUNG AUF BAND

3 - EINLESEN VON DATEN
    AUF BAND

4 - ENDE DER EINGABE

BITTE WAEHLEN
```

```
DATENEINGABE ENDE->0,0

TAG UND MONAT DURCH
KOMMA GETRENNT EIN-
GEBEN.
? 23,4

EINZUTRAGENDE NOTIZ
BIS 50 ZEICHEN
? GEBURTSTAG MAXIMILIA
N

DOPPELEINTRAG FEST-
GESTELLT AM
 23 . 4 . 1984

ALTE EINGABE (1):

OSTERMONTAG

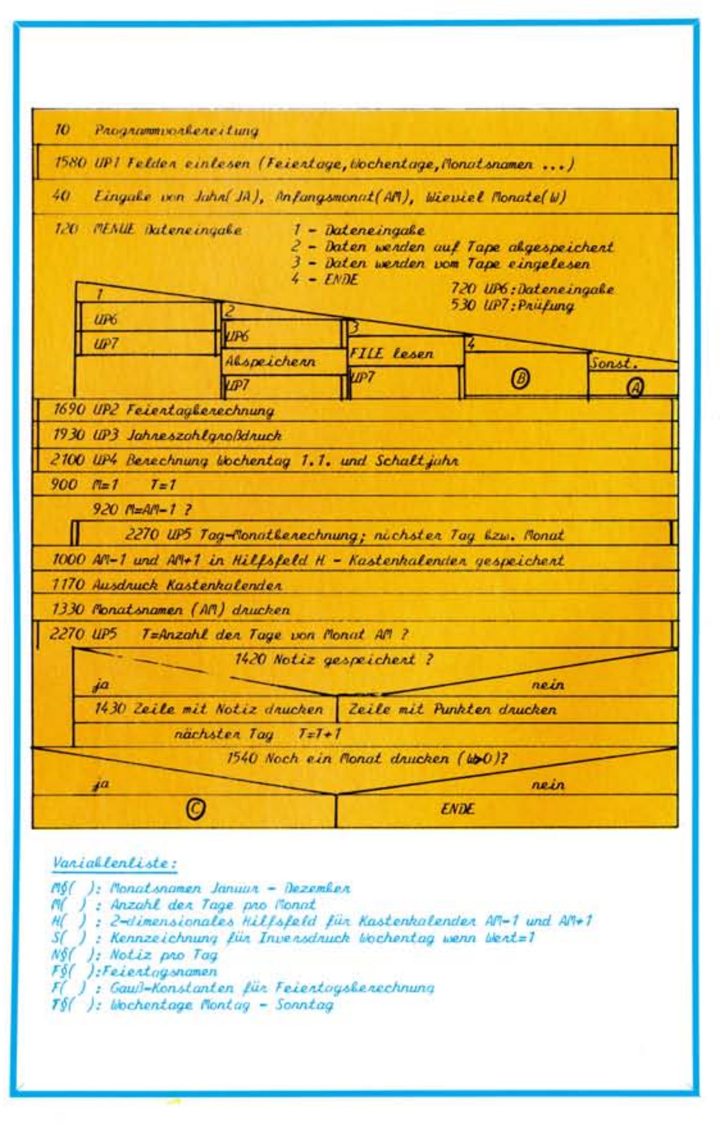
NEUE EINGABE (2):

GEBURTSTAG MAXIMILIAN

WELCHE EINGABE SOLL
UEBERNOMMEN WERDEN 1/2
```

Mit diesen Menüs erfolgt die Dateneingabe für das Kalenderprogramm

Mit dem Programm Kalender können Sie sich für jeden Monat ein individuelles Datenblatt mit allen wichtigen Eintragungen erstellen lassen. Kalender läuft auf einem VC 20 mit 16 KByte Erweiterung. Zur Speicherung dient ein Kassettenrecorder; für den Ausdruck sorgt der Drucker VC 1515.



Nassi-Shneidermann-Diagramm für das Programm »Kalender«. Der Programmablaufplan ist vereinfacht wiedergegeben.

es Notizbuch

Dieses Programm unterscheidet sich von herkömmlichen durch:
— Jahreszahlgroßschrift (aus Zeichengenerator-ROM)
— Zeilen- und Kastenkalenderausdruck des jeweils vorangegangenen und kommenden Monats
— Universelle Bestimmung des Wochentages vom 1.1. zur Bestimmung des Kalenderanfangs
— Berechnung der beweglichen Feiertage
— Eingabe und Ausdruck der nicht beweglichen Feiertage
— Eingabe und Ausdruck sonstiger beliebiger Notizen
— Abspeicherung und Wiedereinlesen beliebiger Notizen
— Möglichkeit des Inversdruckes von Wochentagen und Bemerkungen mittels Steuerkennzeichen
— Leichte Erweiterbarkeit (zum Beispiel Tagesnumerierung, Wochenzählung, Berechnung von Sonnenauf- und Sonnenuntergang...). Alles in allem ein sehr komfortables Kalenderprogramm.

(Jörg Fuhr)

So sind die Unterprogramme aufgebaut

UP1 Felder einlesen

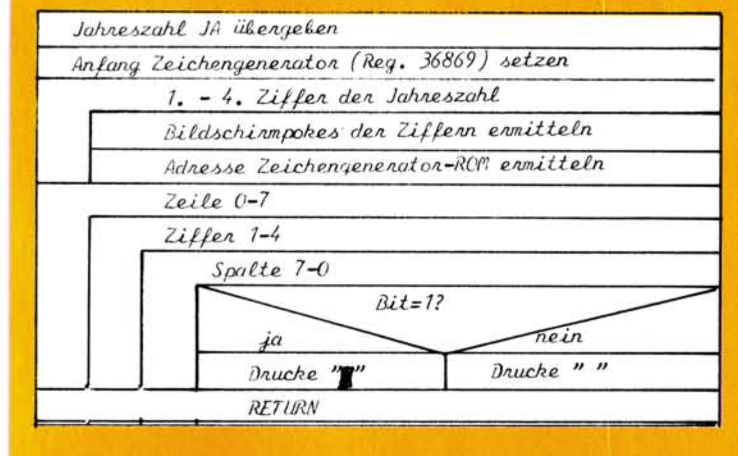
Bewegliche Feiertagsnamen einlesen F$( )
Konstanten für Feiertagsberechnung einlesen F( )
Datum und Notizen einlesen N$(Mx32+T)
Wochentagsnamen einlesen T$( )
Monatsnamen einlesen M$ ( )
Anzahl der Tage pro Monat einlesen M( )
RETURN

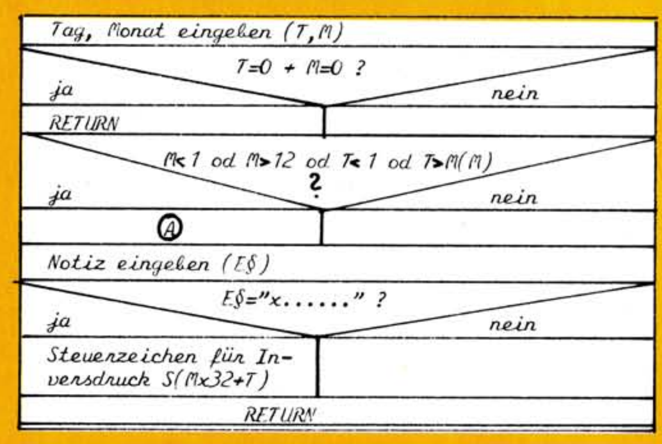
UP2 Bewegliche Feiertage bestimmen

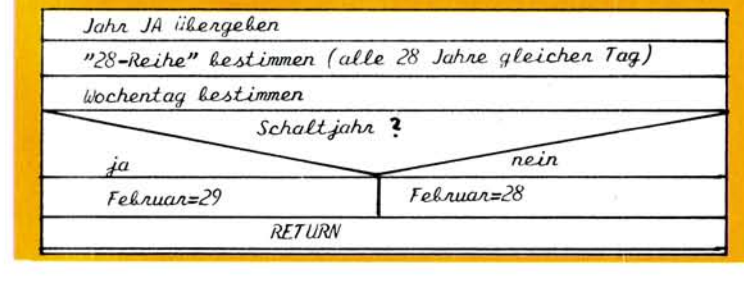
Konstanten bestimmen
Berechnung nach Gauß'scher Formel
Datum und Feiertag in N$(Mx32+T) abspeichern
RETURN

UP5 Tag-Monat-Berechnung

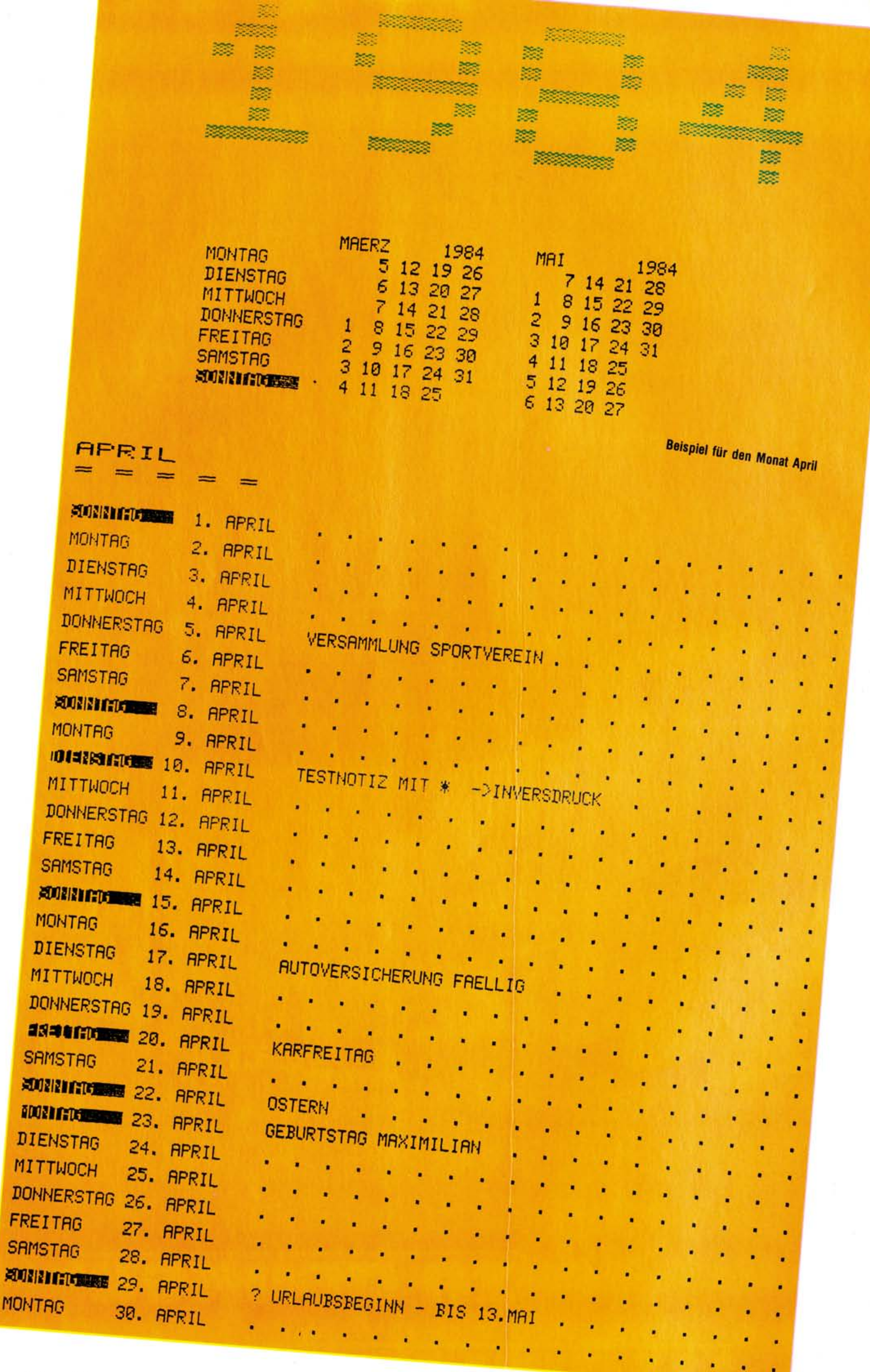
Übergabe Wochentag (WT), Tag (T), Monat (M)	
Tag=Monatsende? ja	nein
T=0 M=M+1	
T=T+1	
WT=WT+1	
WT=8 ? ja	nein
WT=1	
RETURN	











1984

	MAERZ	1984				MAI			1984	
MONTAG		5	12	19	26		7	14	21	28
DIENSTAG		6	13	20	27	1	8	15	22	29
MITTWOCH		7	14	21	28	2	9	16	23	30
DONNERSTAG	1	8	15	22	29	3	10	17	24	31
FREITAG	2	9	16	23	30	4	11	18	25	
SAMSTAG	3	10	17	24	31	5	12	19	26	
SONNTAG	4	11	18	25		6	13	20	27	

APRIL
= = = = =

SONNTAG 1. APRIL
MONTAG 2. APRIL
DIENSTAG 3. APRIL
MITTWOCH 4. APRIL
DONNERSTAG 5. APRIL VERSAMMLUNG SPORTVEREIN
FREITAG 6. APRIL
SAMSTAG 7. APRIL
SONNTAG 8. APRIL
MONTAG 9. APRIL
DIENSTAG 10. APRIL TESTNOTIZ MIT * ->INVERSDRUCK
MITTWOCH 11. APRIL
DONNERSTAG 12. APRIL
FREITAG 13. APRIL
SAMSTAG 14. APRIL
SONNTAG 15. APRIL
MONTAG 16. APRIL
DIENSTAG 17. APRIL AUTOVERSICHERUNG FAELLIG
MITTWOCH 18. APRIL
DONNERSTAG 19. APRIL
FREITAG 20. APRIL KARFREITAG
SAMSTAG 21. APRIL
SONNTAG 22. APRIL OSTERN
MONTAG 23. APRIL GEBURTSTAG MAXIMILIAN
DIENSTAG 24. APRIL
MITTWOCH 25. APRIL
DONNERSTAG 26. APRIL
FREITAG 27. APRIL
SAMSTAG 28. APRIL
SONNTAG 29. APRIL ? URLAUBSBEGINN - BIS 13.MAI
MONTAG 30. APRIL

Beispiel für den Monat April

```
10 REM PROGRAMMVORBEREITUNGEN
20 DIMM$(13),M(12),H(13,7),S(415),N$(415)
30 GOSUB1580
40 PRINT"██ K A L E N D E R":PRINT"██                                    █":PRINT:PRINT:PRINT
50 INPUT"JAHR";JA
60 INPUT"██ANFANGSMONAT";AM
70 PRINT"█ZAHL DER AUSZUDRUCKEN-DEN MONATE"
80 INPUTW
90 W=W-1
100 IFW+AM>12THENPRINT"█KALENDERAUSDRUCK GEHT NUR BIS DEZEMBER"JA:GOTO60
110 GOSUB1690:GOSUB2230
120 REM DATENEINGABE
130 PRINT"██DATENEINGABE█"
140 PRINT"█                █"
150 PRINT"█1 - DATENEINGABE"
160 PRINT"█2 - ZUSAETZLICH SPEI-     CHERUNG AUF BAND"
170 PRINT"█3 - EINLESEN VON DATEN    AUF BAND"
180 PRINT"█4 - ENDE DER EINGABE"
190 PRINT"█BITTE WAEHLEN"
200 GETX$:IFX$=""THEN200
210 IFVAL(X$)<1ORVAL(X$)>5THEN200
220 ONVAL(X$)GOTO230,280,420,840
230 REM DATENEINGABE NUR IN ARBEITSSPEICHER
240 GOSUB720
250 IFFD=1THEN120
260 GOSUB530
270 GOTO230
280 REM DATENEINGABE + ABSPEICHERUNG
290 PRINT"█FILENAME"
300 INPUT NF$
310 OPEN2,1,1
320 PRINT#2,NF$
330 GOSUB720
340 PRINT#2,TM
350 IFFD=1THENCLOSE2:GOTO120
360 PRINT#2,X$+E$
370 GOSUB530
380 IFDE=0THEN410
390 PRINT"NEUE EINGABE WIRD      DENNOCH ABGESPEICHERT!"
400 FORI=1TO3000:NEXT
410 GOTO330
420 REM DATEN VOM BAND EINLESEN
430 PRINT"█FILENAME"
440 INPUTNF$
450 OPEN2,1,0
460 INPUT#2,X$
470 IFNF$<>X$THEN450
480 INPUT#2,TM
490 IFTM=0THENCLOSE2:GOTO120
500 INPUT#2,E$
510 GOSUB530
520 GOTO480
530 REM DOPPELEINTRAG-PRUEFUNG  UP 7
540 DE=0
550 IFN$(TM)=""THENN$(TM)=E$:RETURN
560 M=INT(TM/32):T=TM-32*M
570 PRINT"██DOPPELEINTRAG FEST-   GESTELLT AM"
580 PRINTT"."M"."JA
590 PRINT"█                █"
600 PRINT"█ALTE EINGABE (1):█"
610 PRINTN$(TM)
620 PRINT"██NEUE EINGABE (2):█"
630 PRINTE$
640 PRINT"██WELCHE EINGABE SOLL   UEBERNOMMEN WERDEN 1/2"
650 GETX$:IFX$=""THEN650
660 IFX$="1"THENDE=1:RETURN
670 IFX$="2"THENN$(TM)=E$:RETURN
880 GOTO650
690 REM FEHLERMELDUNG
700 PRINT"████EINGABE- F E H L E R"
710 FORI=1TO3000:NEXT
720 REM DATENEINGABE  UP 6
730 FD=0:X$=""
740 PRINT"██DATENEINGABE ENDE->0,0"
750 PRINT"█                █"
760 PRINT"TAG UND MONAT DURCH     KOMMA GETRENNT EIN-    GEBEN."
770 INPUTT,M:TM=M*32+T
780 IFT=0ANDM=0THENFD=1:E$="":RETURN
790 IFM<1ORM>12ORT<1ORT>M(M)THEN690
800 PRINT"███EINZUTRAGENDE NOTIZ   BIS 50 ZEICHEN"
810 INPUTE$
820 IFLEFT$(E$,1)="*"THENE$=RIGHT$(E$,LEN(E$)-1):S(TM)=1
830 RETURN
840 REM KALENDERAUSDRUCK
```

Listing des Programms »Kalender«

```
850 PRINT"□  W  A  I  T        "
860 POKE36879,8
870 GOSUB1930
880 IFAM=1THENJA=JA-1:AM=13:AJ=1
890 IFAM=12THENZJ=1
900 M=1:T=1
910 GOSUB2100
920 IFM<>AM-1THENGOSUB2270:GOTO920
930 POKE36879,27
940 PRINT"□  DRUCKVORGANG"
950 REM KASTENKALENDER
960 FORX=1TO13
970 FORY=1TO7
980 H(X,Y)=0
990 NEXTY,X
1000 T=1:X=1
1010 FORY=WTTO7
1020 H(X,Y)=T
1030 T=T+1
1040 IFT>M(M)THEN1080
1050 NEXT
1060 WT=1:X=X+1
1070 GOTO1010
1080 IF FLTHEN1170
1090 IFAJTHENJA=JA+1:GOSUB2230
1100 X=X+2:WT=Y+1:M=M+1:TW=WT
1110 IFM=13THENM=1:AM=1
1120 WT=M(M)-7*INT(M(M)/7)+WT
1130 IFWT>7THENWT=WT-7
1140 T=1:M=M+1:FL=1
1150 IFM=13THENM=1
1160 GOTO1010
1170 REM AUSDRUCK KASTENKALENDER
1180 OPEN1,4
1190 PRINT#1," "SPC(23)M$(AM-1)JA-AJ;SPC(4)M$(AM+1)JA+ZJ
1200 FL=0:AJ=0:ZJ=0
1210 FORY=1TO7
1220 PRINT#1," "SPC(10)T$(Y)"  ";
1230 FORX=1TO13
1240 X$=STR$(H(X,Y))
1250 IFX$=" 0"THENX$="  "
1260 IFH(X,Y)<10THENX$=" "+X$
1270 PRINT#1,X$;
1280 NEXTX
1290 PRINT#1
1300 NEXTY
1310 PRINT#1:PRINT#1:PRINT#1:CLOSE1
1320 WT=TW:T=1:M=AM:GOSUB2320
1330 REM MONAT AUSDRUCKEN
1340 OPEN1,4
1350 PRINT#1,CHR$(14)M$(M):PRINT#1,"= = = = =":PRINT#1,CHR$(15)
1360 IFT<10THENTT$=" "+STR$(T)+". "
1370 IFT>9THENTT$=STR$(T)+". "
1380 TH$=T$(WT)
1390 IFS(M*32+T)=1THENTH$="▣"+TH$+"■"
1400 PRINT#1,CHR$(15)TH$;TT$;M$(M);
1410 S$=N$(M*32+T):X=LEN(S$)
1420 IFX=0THENS$=".  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .":GOTO1470
1430 S$=S$+"   ":S$=LEFT$(S$,3+3*INT(X/3)):X=LEN(S$)
1440 FORI=XTO51STEP3
1450 S$=S$+".  "
1460 NEXT
1470 PRINT#1,S$;CHR$(8)
1480 PRINT#1,CHR$(8)
1490 GOSUB2270
1500 IFM<AM+1THEN1360
1510 FORI=M(M-1)TO37
1520 PRINT#1
1530 NEXT
1540 REM WIEDERHOLUNG
1550 CLOSE1
1560 IFW>0THENAM=AM+1:W=W-1:GOTO840
1570 PRINT"□":END
1580 REM FELDER EINLESEN  UP 1
1590 FORI=1TO7:READF$(I):NEXT
1600 FORI=1TO7:READF(I):NEXT
1610 READT,M,E$
1620 IFLEFT$(E$,1)="*"THENE$=RIGHT$(E$,LEN(E$)-1):S(M*32+T)=1
1630 N$(M*32+T)=E$
1640 IFT>0ANDM>0THEN1610
1650 FORI=1TO7:READT$(I):NEXT:REM WOCHENTAGE EINLESEN
1660 FORI=0TO13:READM$(I):NEXT:REM MONATSNAMEN EINLESEN
1670 FORI=1TO12:READM(I):NEXT
1680 RETURN
1690 REM FEIERTAG-BERECHNUNG  UP 2
```

Listing des Programms »Kalender« (Fortsetzung)

```
1700 IFJA<1800THENRETURN
1710 IFJA>=1800THENR=23:S=4
1720 IFJA>=1900THENR=24:S=5
1730 IFJA>=2100THENR=24:S=6
1740 IFJA>=2200THENR=25:S=0
1750 IFJA>=2300THENRETURN
1760 F1=R+19*(JA-19*INT(JA/19))
1770 F=F1-30*INT(F1/30)
1780 G1=JA-4*INT(JA/4)
1790 G2=JA-7*INT(JA/7)
1800 G3=2*G1+4*G2+6*F+S
1810 G=F+G3-7*INT(G3/7)
1820 IFG=35THENG=28
1830 IFG=34ANDF=28ANDJA-19*INT(JA/19))10THENG=27
1840 FORI=1TO7
1850 X=G-F(I)
1860 IFX<=1THENX=X+30.9
1870 IFX<=1THENX=X+29.9
1880 X=320*X-319*INT(X)
1890 X=INT(X+0.5)
1900 N$(X)=F$(I):S(X)=1
1910 NEXTI
1920 RETURN
1930 REM JAHRESZAHLAUSDRUCK  UP 3
1940 ZA=32768:POKEZA,240
1950 OPEN1,4:PRINT#1,CHR$(14);
1960 FORJ=1TO4
1970 Z=VAL(MID$(STR$(JA),J+1,1))
1980 ZZ(J)=ZA+8*(48+Z)
1990 NEXTJ
2000 FORI=0TO7:REM ZEILE
2010 PRINT#1:PRINT#1,"    ";
2020 FORJ=1TO4:REM ZAHL
2030 FORK=7TO0STEP-1:REM SPALTE
2040 IF2↑K AND PEEK(ZZ(J)+I)THENPRINT#1,"■";:GOTO2060
2050 PRINT#1," ";
2060 NEXTK,J,I
2070 PRINT#1,CHR$(15):PRINT#1:PRINT#1:PRINT#1
2080 CLOSE1
2090 RETURN
2100 REM BERECHNUNG WOCHENTAG 1.JANUAR  UP 4
2110 X1=JA-28*INT(JA/28)
2120 IFX1=0THENX1=28
2130 IFX1>14.5THENX2=X1-14:GOTO2150
2140 ONX1GOTO2220,2160,2170,2180,2200,2210,2220,2160,2180,2190,2200,2210,2160,21
70
2150 ONX2GOTO2180,2190,2210,2220,2160,2170,2190,2200,2210,2220,2170,2180,2190,22
00
2160 WT=1:GOTO2230
2170 WT=2:GOTO2230
2180 WT=3:GOTO2230
2190 WT=4:GOTO2230
2200 WT=5:GOTO2230
2210 WT=6:GOTO2230
2220 WT=7:GOTO2230
2230 REM SCHALTJAHRBERECHNUNG
2240 M(2)=28
2250 IFJA/4<>INT(JA/4)THENRETURN
2255 IFJA/100<>INT(JA/100)OR(JA/100=INT(JA/100)ANDJA/400=INT(JA/400))THENM(2)=M(
2)+1
2260 RETURN
2270 REM TAG-MONAT-BERECHNUNG  UP 5
2280 REM WT+T AUS UP
2290 IFT=M(M)THENT=0:M=M+1
2300 T=T+1
2310 WT=WT+1
2320 IFWT=8THENWT=1
2330 RETURN
2340 DATAOSTERN,KARFREITAG,OSTERMONTAG,HIMMELFAHRT,PFINGSTEN,PFINGSTMONTAG,FRONL
EICHNAM
2350 DATA8.6,10.6,7.6,30.4,20.4,19.4,9.4
2360 DATA1,1,*NEUJAHR,1,5,*MAIFEIERTAG,17,6,*GESETZL.FEIERTAG,1,11,*ALLERHEILIGE
N
2370 DATA24,12,HEILIG ABEND,25,12,*1.WEIHNACHTSTAG,26,12,*2.WEIHNACHTSTAG
2380 DATA21,3,FRUEHLINGSANFANG,22,6,SOMMERANFANG,23,9,HERBSTANFANG
2390 DATA22,12,WINTERANFANG,0,0,
2400 DATAMONTAG    ,DIENSTAG  ,MITTWOCH  ,DONNERSTAG
2410 DATA FREITAG   ,SAMSTAG  ,"■SONNTAG  ■"
2420 DATA DEZEMBER ,JANUAR   ,FEBRUAR  ,MAERZ    ,APRIL    ,MAI      ,"JUNI
"
2430 DATA JULI     ,AUGUST   ,SEPTEMBER,OKTOBER  ,NOVEMBER ,DEZEMBER ,"JANUAR
"
2440 DATA 31,28,31,30,31,30,31,31,30,31,30,31
READY.
```

Listing des Programms »Kalender« (Schluß)

COMP
UND S

Dieses Programm wurde zur Auswertung von Deutschen Mannschafts-Meisterschaften entwickelt. Denkbar ist aber auch, daß man es bei der Vereinsmeisterschaft im eigenen Ort einsetzt. Notwendig ist dabei nur der Commodore 64 mit Peripherie.



▲ Bild 1. Hauptmenü. Rechts oben in der Ecke erscheint die Uhrzeit. Das Anwählen der einzelnen Unterprogramme erfolgt über Druck auf die entsprechende Zahlentaste

```
BUNDESLIGA MAENNER
1  TSG 78 HEIDELBERG
2  TSV SCHOENAU
3  VFB GAGGENAU
4  LAG OBERE MURG
5  LSG SAARBR.SULZBACHTAL

Z  ZWISCHENERGEBNIS
A  ANFANG
K  KONTROLLE
GEBEN SIE DIE NR.DES VEREINS EIN
?
```

▲ Bild 2. Untermenü. Die Gruppe 1 wurde angewählt. Oben erscheint die Liste der Mannschaften, darunter die anderen Programmöglichkeiten

```
1  100M
2  200M
3  400M
4  800M
5  1500M
6  5000M
7  110M H
8  400M H
9  4X100M
10 HOCH
11 STABHOCH
12 WEIT
13 DREI
14 KUGEL
15 DISKUS
16 HAMMER
17 SPEER
GEBEN SIE NR. DER DISZIPLIN (0)=AUSSTIEG
?
```

▲ Bild 3. Erste Eingabe. Die Disziplin muß eingegegeben werden. Hier gelangt man hin, wenn man eine Mannschaft angewählt hat

Meine Arbeit mit dem Computer begann im Juli 1982. Ich war in der 12. Klasse und besuchte einen Informatik Grundkurs. Die dort erworbenen, jedoch relativ geringen Grundlagen der Basic-Programmierung führten durch eigene Versuche an Spielen und an einem NGO-Verwaltungsprogramm, im Teamwork, relativ schnell zu einer guten Kenntnis der Basic-Programmiersprache und teilweise auch schon zu Maschinensprachekenntnissen. Diese wurden durch den Kauf eines C 64 im April 1983 ausgebaut und bildeten so die Basis zur Erstellung sinnvoller Programme. Der Einsatz des Computers im Sport lag also nicht fern, zumal mein Vater Leichtathletiksportfeste organisiert. So war er es auch, der mich fragte, ob der Computer bei einem DMM-Wettkampf nicht gute Dienste leisten könnte. Die Programmidee war damit geboren.

Eine Idee, eine Herausforderung und der Commodore 64

Die nähere Problematik soll nun für alle erläutert werden, die mit dem Ablauf von Leichtathletikwettkämpfen weniger vertraut sind. Das Kürzel DMM steht für Deutsche Mannschafts-Meisterschaften. Diese werden in verschiedenen Leistungsklassen und auf unterschiedlichen Ebenen durchgeführt. Bei dem Wettkampf, für den das Programm ursprünglich geschrieben wurde, handelte es sich um den Endkampf um die badische Mannschaftsmeisterschaft für Männer und Frauen in jeweils drei Klassen. Es waren Bezirksliga, Landesliga und Bundesliga vertreten. Diese drei Gruppen unterscheiden sich untereinander durch eine unterschiedliche Anzahl an Disziplinen, wobei die Disziplinen bei den Männern sich von denen bei den Frauen nochmals unterscheiden. (Man denke beispielsweise an die unterschiedlichen Gewichte beim Kugelstoßen etc.)

Ein MByte für das DLV-Punktebuch

Jede der teilnehmenden Mannschaften, deren Anzahl in den einzelnen Gruppen nochmals recht unterschiedlich ausfällt, kann nun pro Disziplin maximal drei Teilnehmer stellen. Die Ergebnisse, die von diesen erzielt werden, werden an-

UTER PORT

```
EINGABE TSG 78 HEIDELBERG
DISZIPLIN:  1  100M

ERGEBNIS NR.: 1
? 564

ERGEBNIS NR.: 2
? 546

ERGEBNIS NR.: 3
? 567
```

Bild 4. Nach Eingabe der Disziplin erfolgt die Eingabe der drei Einzelleistungen. In diesem Fall bei Mannschaft 1 der Gruppe 1

Bild 5. Die eingegebenen Ergebnisse werden sortiert und ausgegeben. Das rote Ergebnis ist nicht in die Wertung gekommen. Darunter erscheint der Gesamtpunktestand. Rechts oben in der Ecke läuft wieder die Zeit

```
SORTIERTE ERGEBNISSE:              15:00:02

 567
 564

 546

TOTAL TSG 78 HEIDELBERG: 1131

TASTE DRUECKEN
```

```
ZWISCHENERGEBNIS:                  15:00:47
BUNDESLIGA MAENNER

    1230 LSG SAARBR.SULZBACHTAL
    1131 TSG 78 HEIDELBERG
    1085 TSV SCHOENAU
    1067 VFB GAGGENAU
    1021 LAG OBERE MURG

STAND NACH DER ERSTEN DISZIPLIN
TASTE DRUECKEN!
(D FUER AUSGABE AUF DRUCKER)

AUSGEFUEHRTE DISZIPLINEN:
100M
```

Bild 6. Vom Untermenü (Bild 2) wurde in das Zwischenergebnis verzweigt. Die Mannschaften sind nach Punktezahlen geordnet. Unten sind die Disziplinen aufgeführt, die bereits von allen Mannschaften komplett absolviert wurden. Rechts oben läuft wieder die Zeit. Wenn jetzt die »D«-Taste gedrückt wird, erfolgt der bereits Ihnen zugeleitete Ausdruck

hand eines DLV-Punktebuches in die entsprechenden Punktezahlen umgesetzt. Da diese Punkte leider recht wahllos und nicht durch irgendeine Formel bestimmbar sind, ist an eine Auswertung mit dem Computer an dieser Stelle noch nicht zu denken. Das DLV-Punktebuch enthält nämlich zirka 83000 unterschiedliche Leistungen und zugeordnete Punktezahlen. Der Speicheraufwand in Basic läge also schätzungsweise bei einem MByte, was den Speicherbereich des C 64 selbst mit angeschlossenen Floppy-Laufwerken erheblich übersteigt. Also mußte ich mir etwas anderes überlegen: Der Rechenvorgang, der, nachdem die Punkte festgestellt sind, vollzogen werden muß, ist jedoch sehr gut mit einem Computer zu realisieren. Die drei Teilergebnisse einer Mannschaft in einer Disziplin müssen sortiert werden. Die beiden besten Ergebnisse werden nun zum Gesamtergebnis hinzuaddiert. Dies ist wohl ein Vorgang, dessen Einfachheit besticht, der aber gleichzeitig die Frage aufwirft, wozu dann ein Programm benötigt wird. Die Antwort auf diese Frage ist recht einfach. Es geht um die Geschwindigkeit und die Zuverlässigkeit. Hinzu kommt noch die Möglichkeit einer schnellen Gegenkontrolle, da alle Einzelergebnisse gespeichert bleiben und schnell abrufbar sind. Das Wichtigste ist jedoch das schnelle Herstellen von Zwischenergebnissen, wozu im Wettkampfbüro keine Zeit vorhanden ist. Diese Zwischenergebnisse können nun ohne großen Aufwand, nur durch wenige Tastendrücke zu jedem beliebigen Zeitpunkt abgerufen werden.

Im Folgenden soll nun beschrieben werden, wie der Ablauf in der Praxis vor sich ging. Die Wettkampfkarten mit den Ergebnissen kamen ins Wettkampfbüro und dort wurden dann die entsprechenden Punkteergebnisse herausgesucht. Der Zettel mit den Punkteergebnissen kam zum Computer und die Ergebnisse wurden eingegeben. Der Computer wählte die beiden besten Ergebnisse einer Mannschaft aus und addierte sie zum Gesamtergebnis. War eine Disziplin einer Gruppe abgeschlossen, so wurde das Zwischenergebnis mit Uhrzeit ausgedruckt und dem Stadionsprecher weitergeleitet, um diesem immer die aktuellsten Informationen zu sichern. Danach ging eine Kopie der Zwischenergebnisse an das Wettkampfbüro zurück, das dann eine Vergleichsmöglichkeit hatte, wenn es später zu seinem handgerechneten Zwischenergebnis kam. Dies war immer erst wesentlich später der Fall, da die Ergebnisse immer noch in mehrere andere Listen übertragen werden mußten, bevor sie addiert wurden. Die Möglichkeiten einer schnellen Kontrolle aller Einzelergebnisse durch den Computer, ohne einen Berg von Wettkampfkarten durchsuchen zu müssen, erwies sich als äußerst praktisch bei der Fehlersuche. Die Fehler, die auftraten und durch das Zwischenergebnis des Computers aufgedeckt wurden,

waren meist Schreib- oder Lesefehler, Übertragungsfehler von einer Liste auf die nächste oder schlichte Rechenfehler, die sich leicht einschleichen, wenn man in der Hektik eines Wettkampfbüros auch noch schnell per Kopf oder Taschenrechner rechnen soll.

Der Computer hilft Fehler vermeiden

So war die Fehlerquelle mit Hilfe des Computers schnell entdeckt und der Fehler wurde beseitigt. Ein Schritt zu einer besseren und fehlerfreien Ergebnisliste, die sich sehen lassen kann.

Das Programm selbst entstand in zirka fünfstündiger Arbeit an einem Sonntagnachmittag und wurde, nach seinem Einsatz, den praktischen Erfahrungen gemäß, die damit gemacht wurden, erweitert und verbessert, es gibt schließlich viele Dinge, auf die man rein theoretisch niemals kommt, die in der Praxis aber wünschenswert sind. Der Peripheriemangel, der zur Entstehungszeit des Programmes herrschte ist dafür verantwortlich, daß im ursprünglichen Programm die Datenspeicherung nicht gleich auf Diskette erfolgte. Diese Ergänzungen kamen erst später hinzu. Das hatte jedoch auch einen guten Nebeneffekt, nämlich den, daß das Programm mit praktisch allen Peripheriekonfigurationen sinnvoll arbeiten kann. Das beginnt bei Kassettenstation und Zentraleinheit und endet bei Zentraleinheit, Floppy und Drucker. Die einzelnen Bildschirmmasken nutzen die farblichen Möglichkeiten des C 64 zur übersichtlichen Darstellung der Daten. POKE-Befehle wurden weitgehend vermieden, um das Programm möglichst leicht auch auf andere Computer übertragen zu können. POKE-Befehle wurden nur zur Erzeugung der Hintergrundfarben und des Tones verwendet und können an diesen Stellen des Programmes durchaus entfernt werden. Auf die Verwendung von REM-Zeilen wurde verzichtet, da der durch sie verbrauchte, unnötige Speicherplatz das Programm verlangsamt und unnötig verlängert. Zusätzlich würde die verwendbare Datenmenge auch noch eingeschränkt.

Das Program ist im Menü-System aufgebaut, so daß es recht einfach benutzt werden kann. Die einzelnen Möglichkeiten, die der Benutzer jetzt hat sind meist alle auf dem Bildschirm dargestellt. Das Programm soll

BUNDESLIGA MAENNER
VEREIN:TSG 78 HEIDELBERG
DISZIPLIN:100M
ERGEBNIS 1 : 567
ERGEBNIS 2 : 564
ERGEBNIS 3 : 546
TASTE DRUECKEN
NR. DES ERGEBNISSES FUER KORREKTUR

▲ Bild 7. Der Unterprogrammschritt Kontrolle wurde vom Untermenü (Bild 2) aus angewählt. Wird die Taste 1, 2 oder 3 gedrückt, dann kann das entsprechende Ergebnis korrigiert werden



▲ Bild 8. Ein Zwischenergebnis der Gruppe 3. Es wurden mehr Disziplinen ausgeführt

nun einmal von Anfang an theoretisch durchlaufen werden. Wird das Programm gestartet, so erfolgt als Erstes die Eingabe der Zeit. Wurde das Programm vorher bereits einmal gestartet, so kann einfach nur »RETURN« eingegeben werden. Die vorher eingegebene Zeit bleibt dann erhalten, da die interne Computeruhr, die ja ständig mitläuft, vom Programm aus benutzt wird. Die Eingabe der Zeit erfolgt im »HHMMSS«-Format. Das bedeutet, daß zum Beispiel 12 Uhr 23 und 15 Sekunden als 122315 eingegeben wird. Nach dieser Eingabe befindet man sich im Hauptmenü (Bild 1), das zunächst sechs gleichwertige Untermenüs anbietet, die den einzelnen Gruppen Landesliga, Bundesliga und Bezirksliga Männer bzw. Frauen zugeordnet sind. Diese Untermenüs sind alle auf dem gleichen Unterprogramm aufgebaut. Der Unterschied liegt nur in den Daten, also den Namen der Mannschaften, der Anzahl der Disziplinen und der Art der Disziplinen. Wird nun eines der Untermenüs (Bild 2), angewählt, so erscheint eine Liste der Mannschaften, mit der jeweils fortlaufenden Mannschaftsnummer. Das Untermenü bietet nun die Möglichkeit der Dateneingabe bei den einzelnen Mannschaften, der Kontrolle aller Einzelergebnisse einer Mannschaft, der Ausgabe der Zwischenergebnisse und des Rücksprunges in das Hauptmenü. Für den Aussprung zurück ins Hauptmenü gibt man »A« ein. Zu den Eingaben ist grundsätzlich noch folgendes zu sagen: Blinkt der Cursor so muß nach der Eingabe »RETURN« gedrückt werden, ist er jedoch nicht zu sehen, so ist dies nicht notwendig.

Menütechnik sorgt für Übersichtlichkeit

Entscheidet man sich in diesem Untermenü für die Dateneingabe, so muß einfach die Nummer der Mannschaft eingegeben werden, bei der Daten eingegeben werden sollen. Darauf erscheint eine Liste der in dieser Gruppe vorhandenen Disziplinen (Bild 3), geordnet nach Lauf, Sprung und Wurf. Wird nun die Nummer der Disziplin eingegeben, so überprüft das Programm zunächst, ob diese Ergebnisse nicht vielleicht schon eingegeben wurden.

Sicherung gegen Fehleingabe ist ein Muß

Ist dies der Fall, so wird, um eine fehlerhafte Eingabe zu verhindern auf dem Bildschirm eine entsprechende Fehlermeldung ausgegeben. Diese wird von einem akustischen Signal begleitet, da man die Meldung auf dem Bildschirm leicht übersieht, wenn man sich auf die Tastatur konzentriert. Das Programm versichert sich nun, ob die Daten wirklich neu eingegeben werden

sollen. Dies geschieht durch eine einfache Ja/Nein-Abfrage. So ist gewährleistet, daß versehentlich gemachte Eingaben auf einfachstem Wege immer wieder gelöscht werden können, daß auf der anderen Seite aber die bereits vorhandenen Daten geschützt sind. Gibt man nun die Punkteergebnisse ein (Bild 4) und drückt nur »RETURN«, so nimmt das Programm als Punktezahl nur einfach eine Null an. Die Punkteergebnisse werden nun sortiert und auf dem Bildschirm übersichtlich ausgegeben. Die beiden besten Ergebnisse stehen oben und sind grün gefärbt (Bild 5), weil sie zum Gesamtergebnis addiert werden. Das dritte Ergebnis wird etwas darunter rot eingefärbt ausgegeben, da es nicht in die Wertung kommt.

Alle Ergebnisse werden zur Kontrolle gespeichert

Dieses Ergebnis wird jedoch trotzdem so wie die beiden anderen gespeichert, da es bei Rückfragen zur Kontrolle manchmal doch eine Rolle spielt. Ist die Dateneingabe abgeschlossen, kehrt man wieder in das Untermenü mit der Liste der Mannschaften zurück.

Wird in diesem Untermenü »Z« für Zwischenergebnisse eingegeben, so erscheint eine Liste der Mannschaften in Reihenfolge der von ihnen erzielten Punktezahlen (Bild 6). Ferner erscheint eine Liste der von ihnen bereits durchgeführten Disziplinen. Hier spielt nun auch die beim Start des Programmes eingegebene Zeit eine Rolle, da sie auf dem Ausdruck mit angegeben wird. Die Zeit, zu der ein bestimmtes Zwischenergebnis vorlag ist rein statistisch interessant und kann später zur Optimierung des Zeitplanes verwendet werden. Unter der Liste der Mannschaften wird noch angegeben wieviele Disziplinen bereits ausgeführt sind, wobei nur Disziplinen gezählt werden, die alle Mannschaften bereits durchgeführt haben. Diese Meldung ist nicht nur stur errechnet, das heißt, daß nicht etwa auf dem Bildschirm erscheint: »Stand nach 0 Disziplinen« oder »Stand nach 1 Disziplinen«.

»Kleinigkeiten« werden beachtet

In diesen beiden Sonderfällen wird stattdessen »Noch keine Disziplin komplett« bzw. »Stand nach der ersten Disziplin« ausgegeben. Zurück ins Untermenü kommt man nun wenn eine beliebige Taste gedrückt wird. Soll die Liste vorher noch auf dem Drucker ausgegeben werden, so drückt man einfach »D«. Nach dem Druckvorgang kommt man ebenfalls in das Untermenü mit der Liste der Mannschaften.

Wird in diesem Untermenü »K« für Kontrolle gedrückt, so erfolgt der Zugriff auf die konkreten Daten wieder über die Vereinsnummer und die Nummer der Disziplin. Auf dem Bildschirm erscheinen dann Gruppe, Verein, Disziplin und die drei Ergebnisse (Bild 7). Das Programm wartet nun wieder auf einen Tastendruck. Wird entweder 1, 2 oder 3 gedrückt, so kann das entsprechende Ergebnis neu eingegeben werden. Bei anderen Tasten erfolgt ein Rücksprung ins Untermenü.

Soll korrigiert werden, so werden die alten Ergebnisse vom Gesamtergebnis subtrahiert. Das neue Ergebnis wird eingegeben, die Ergebnisse werden sortiert und auf dem Bildschirm ausgegeben. Die beiden besten Ergebnisse werden nun wieder zum Gesamtergebnis hinzuaddiert.

Korrekturen ohne weiteres möglich

Diese Untermenümöglichkeiten bestehen für alle sechs Gruppen. Ein Ausstieg aus dem gewählten Programmschritt ist meistens möglich, falls er einmal unbeabsichtigt aufgerufen wird. Die Eingaben werden, so wie das möglich ist, auf Korrektheit überprüft. Im Falle einer festgestellten Unkorrektheit wird der Cur-

```
ZWISCHENERGEBNIS:                    15:03:10
BEZIRKSLIGA FRAUEN

      0 USC FREIBURG
      0 TS OTTERSDORF
      0 TG OETIGHEIM
      0 LG COOP KURPFALZ
      0 ---------

NOCH KEINE DISZIPLIN KOMPLETT!
TASTE DRUECKEN!
(D FUER AUSGABE AUF DRUCKER)

AUSGEFUEHRTE DISZIPLINEN:
```

◄ Bild 9. Ein Zwischenergebnis der Gruppe 6. Ohne bisherige Eingaben

```
BITTE DIE KASSETTE FERTIGMACHEN

UND DANN TASTE DRUECKEN

E-FUER AUSSTIEG EINGEBEN

D-FUER DISKETTE EINGEBEN
```

Bild 10. Die Bildschirmmaske, die erscheint, wenn die Daten auf einen anderen Datenträger übertragen werden sollen. Kassette und Diskette sind möglich ►

```
BUNDESLIGA MAENNER

VEREIN:TSG 78 HEIDELBERG

DISZIPLIN:100M

ERGEBNIS 1  : 567
ERGEBNIS 2  : 564
ERGEBNIS 3  : 546

TASTE DRUECKEN

NR. DES ERGEBNISSES FUER KORREKTUR
NEUES ERGEBNIS 1 ? 590
```

▲ Bild 11. Nochmals Korrekturunterprogramm. Die Eingabe Nr. 1 wurde geändert

```
100 POKE53281,0:POKE53280,0:PRINT"          D M M - P R O G R A M M"
110 PRINT"          COPYRIGHTS BY
120 PRINT"          VOLKER RITZHAUPT
130 PRINT"              1983":WAIT203,63:GETX$:A=27:B=17
140 PRINT"":POKE53281,6:POKE53280,6:GOSUB850:INPUT"ZEIT (FORMAT:HHMMSS)";TI$
150 DATATSG 78 HEIDELBERG,TSV SCHOENAU,VFB GAGGENAU
160 DATALAG OBERE MURG,LSG SAARBR.SULZBACHTAL
170 DATATSG AKUS WEINHEIM,STV SINGEN,TV LAHR,TV GENGENBACH,TV RADOLPHZELL
180 DATA---------,TG OETIGHEIM,USC FREIBURG,TV ZUNSWEIER
190 DATAUSC HEIDELBERG,LAV RALA LUDWIGSHAFEN
200 DATATV LAHR,TSG AKUS WEINHEIM,TV EPPELHEIM,TSV MANNHEIM,SV WALDKIRCH
210 DATA---------,TG OETIGHEIM,---------,USC FREIBURG,TS OTTERSDORF
220 DATALG COOP KURPFALZ
230 DIMA$(A),A(A),B$(B),C(A,B,3),E$(A),E(A,B)
240 FORI=1TOA:READA$(I):NEXT:FORI=1TOA:E$(I)="0":NEXT:C$="":F$=""
250 PRINT"        WELCHE GRUPPE ?
260 G$(1)="BUNDESLIGA MAENNER
270 PRINT"   1 "G$(1)
280 G$(2)="LANDESLIGA MAENNER
290 PRINT"   2 "G$(2)
300 G$(3)="BEZIRKSLIGA MAENNER
310 PRINT"   3 "G$(3)
320 G$(4)="BUNDESLIGA FRAUEN
330 PRINT"   4 "G$(4)
340 G$(5)="LANDESLIGA FRAUEN
350 PRINT"   5 "G$(5)
360 G$(6)="BEZIRKSLIGA FRAUEN
370 PRINT"   6 "G$(6)
380 PRINT"   7 ABSPEICHERN":PRINT"   8 EINLESEN
390 PRINT"   9 PROGRAMMENDE
400 GOSUB1650
410 X1=VAL(X$):IFX1<1THEN400
420 IFX1=9THENPRINT"":END
430 ONX1GOSUB1050,1110,1170,1220,1290,1350,1510,1580
440 IFP0=1THENP0=0:GOTO250
450 X=0:D=0:PRINT"  "G$(X1):PRINT"":FORI=FTOG:PRINTI" "A$(I):NEXT
460 PRINT" Z  ZWISCHENERGEBNIS":PRINT" A  ANFANG":PRINT" K  KONTROLLE
470 PRINT"GEBEN SIE DIE NR.DES VEREINS EIN":D$=""
480 INPUTX$:IFX$="A"THEN250
490 IFX$="Z"THEN700
500 IFX$="K"THENK0=1:PRINT"VEREINSNUMMER ZUR KONTROLLE":GOTO480
510 X=VAL(X$):IFX<F ORX>GTHENPRINT"":GOTO480
520 PRINT"":FORI=1TOH:PRINTB$(I):NEXT:PRINT"GEBEN SIE NR. DER DISZIPLIN";
530 PRINT" (0)=AUSSTIEG"
540 INPUTD$:IFD$=""ORVAL(D$)<0ORVAL(D$)>HTHENPRINT"":GOTO540
550 D=VAL(D$):IFD=0THEN450
560 IFK0=1THENK0=0:GOTO950
570 IFE(X,D)=1THEN870
580 E(X,D)=1
590 PRINT"EINGABE "A$(X)
600 PRINT"DISZIPLIN: "B$(D)
610 PRINT:FORI=1TO3:PRINT"ERGEBNIS NR. :"I:INPUTC(X,D,I):NEXT
620 FORI=1TO2:FORJ=I+1TO3:IFC(X,D,J)<=C(X,D,I)THEN640
630 RI=C(X,D,I):C(X,D,I)=C(X,D,J):C(X,D,J)=RI
640 NEXTJ,I
650 PRINT"SORTIERTE ERGEBNISSE:":PRINTC(X,D,1):PRINTC(X,D,2):PRINT""
660 PRINTC(X,D,3):A(X)=A(X)+C(X,D,1)+C(X,D,2)
670 PRINT"TOTAL "A$(X)":"A(X):PRINT"TASTE DRUECKEN"
680 GOSUB1650
690 GOTO450
700 GOSUB830:FORI=FTOG-1:FORJ=I+1TOG:IFE$(J)<=E$(I)THEN720
710 RI$=E$(I):E$(I)=E$(J):E$(J)=RI$
720 NEXTJ,I:PRINT""C$C$C$C$"AUSGEFUEHRTE DISZIPLINEN:"
730 V=0:FORI=1TOB:FORJ=FTOG:IFE(J,I)<>0THENV1=V1+1
740 NEXTJ:IFV1=G1THENPRINTRIGHT$(B$(I),LEN(B$(I))-4),:V=V+1
750 V1=0:NEXTI
760 PRINT""C$:FORI=FTOG:PRINTE$(I):NEXT:PRINT""
770 IFV=1THENPRINT"STAND NACH DER ERSTEN DISZIPLIN":GOTO800
780 IFV=0THENPRINT"NOCH KEINE DISZIPLIN KOMPLETT!":GOTO800
790 PRINT"STAND NACH"V" DISZIPLINEN!"
800 PRINT"TASTE DRUECKEN!":PRINT"(D FUER AUSGABE AUF DRUCKER)
810 GOSUB1650:IFX$="D"THENGOSUB3000
820 GOTO450
830 PRINT"ZWISCHENERGEBNIS:":PRINT""G$(X1):FORI=FTOG
840 E1$="       "+STR$(A(I)):E$(I)=RIGHT$(E1$,7)+" "+A$(I):NEXT:RETURN
850 S1=54272:POKE54296,15
860 POKES1+3,8:POKES1+6,240:T=64:RETURN
870 PRINT""C$C$"        WURDE SCHON EINGEGEBEN!!!"
880 POKES1+1,40:POKES1,80:FORI=1TO5:POKES1+4,T+1:FORJ=1TO50:NEXT
890 POKES1+4,T:FORJ=1TO30:NEXTJ,I:POKES1+1,0:POKES1,0
900 PRINT"      WOLLEN SIE NEU EINGEBEN? (J/N)"
910 GETY$:IFY$=""THEN910
920 IFY$="J"THEN940
930 GOTO450
940 FORI=1TO2:A(X)=A(X)-C(X,D,I):NEXT:GOTO580
950 PRINT""G$(X1):PRINT"VEREIN:"A$(X)
960 PRINT"DISZIPLIN:"RIGHT$(B$(D),LEN(B$(D))-4)
970 PRINT:FORI=1TO3:PRINT"ERGEBNIS"I" :"C(X,D,I):NEXT
980 PRINT"TASTE DRUECKEN"
990 PRINT"NR. DES ERGEBNISSES FUER KORREKTUR"
1000 GETX$:IFX$=""THEN1000
1010 IFX$="1"ORX$="2"ORX$="3"THEN1030
1020 GOTO450
```

Listing des DMM-Programms

```
1030 I=VAL(X$):A(X)=A(X)-C(X,D,1)-C(X,D,2)
1040 PRINT"NEUES ERGEBNIS"I;:INPUTC(X,D,I):GOTO620
1050 H=B:F=1:G=5:G1=G-F+1:B$(1)=" 1  100M":B$(2)=" 2  200M":B$(3)=" 3  400M"
1060 B$(4)=" 4  800M":B$(5)=" 5  1500M":B$(6)=" 6  5000M"
1070 B$(7)=" 7  110M H":B$(8)=" 8  400M H":B$(9)=" 9  4X100M"
1080 B$(10)="10  HOCH":B$(11)="11  STABHOCH":B$(12)="12  WEIT"
1090 B$(13)="13  DREI":B$(14)="14  KUGEL":B$(15)="15  DISKUS"
1100 B$(16)="16  HAMMER":B$(17)="17  SPEER":RETURN
1110 H=9:F=6:G=10:G1=G-F+1:B$(1)=" 1  100M":B$(2)=" 2  400M"
1120 B$(3)=" 3  800M":B$(4)=" 4  5000M"
1130 B$(5)=" 5  4X100M"
1140 B$(6)=" 6  HOCH":B$(7)=" 7  WEIT"
1150 B$(8)=" 8  KUGEL":B$(9)=" 9  DISKUS"
1160 RETURN
1170 H=4:F=11:G=14:G1=G-F+1:B$(1)=" 1  100M"
1180 B$(2)=" 2  4X100M"
1190 B$(3)=" 3  WEIT"
1200 B$(4)=" 4  KUGEL"
1210 RETURN
1220 H=12:F=15:G=16:G1=G-F+1:B$(1)=" 1  100M":B$(2)=" 2  200M":B$(3)=" 3  400M"
1230 B$(4)=" 4  800M":B$(5)=" 5  3000M"
1240 B$(6)=" 6  100M H":B$(7)=" 7  4X100M"
1250 B$(8)=" 8  HOCH":B$(9)=" 9  WEIT"
1260 B$(10)="10  KUGEL":B$(11)="11  DISKUS"
1270 B$(12)="12  SPEER"
1280 RETURN
1290 H=7:F=17:G=22:G1=G-F+1:B$(1)=" 1  100M"
1300 B$(2)=" 2  800M"
1310 B$(3)=" 3  4X100M"
1320 B$(4)=" 4  HOCH":B$(5)=" 5  WEIT"
1330 B$(6)=" 6  KUGEL":B$(7)=" 7  DISKUS"
1340 RETURN
1350 H=4:F=23:G=27:G1=G-F+1:B$(1)=" 1  100M"
1360 B$(2)=" 2  4X100M"
1370 B$(3)=" 3  WEIT"
1380 B$(4)=" 4  KUGEL"
1390 RETURN
1510 GOSUB1680:IFP$="E"THEN250
1520 IFP$="D"THEN1740
1530 OPEN1,1,1,"DATEN
1540 FORI=1TOA:PRINT#1,A(I):PRINTI,:NEXT
1550 FORI=1TOA:FORJ=1TOB:FORK=1TO3:PRINT#1,C(I,J,K):PRINTI,:NEXTK,J,I
1560 FORI=1TOA:FORJ=1TOB:PRINT#1,E(I,J):PRINTI,:NEXTJ,I
1570 CLOSE1:P0=1:RETURN
1580 GOSUB1680:IFP$="E"THEN250
1590 IFP$="D"THEN1830
1600 OPEN1,1,0,"DATEN
1610 FORI=1TOA:PRINTI,:INPUT#1,A(I):NEXT
1620 FORI=1TOA:FORJ=1TOB:FORK=1TO3:INPUT#1,C(I,J,K):PRINTI,:NEXTK,J,I
1630 FORI=1TOA:FORJ=1TOB:INPUT#1,E(I,J):PRINTI,:NEXTJ,I
1640 CLOSE1:P0=1:RETURN
1650 GETX$:T1$=LEFT$(TI$,2)+":":T2$=MID$(TI$,3,2)+":":T3$=RIGHT$(TI$,2)
1660 PRINT"[illegible]"TAB(30)T1$T2$T3$:IFX$=""THEN1650
1670 RETURN
1680 PRINTF$"[illegible]BITTE DIE KASSETTE FERTIGMACHEN
1690 PRINT"[illegible]UND DANN TASTE DRUECKEN
1700 PRINT"[illegible]E-FUER AUSSTIEG EINGEBEN
1710 PRINT"[illegible]D-FUER DISKETTE EINGEBEN
1720 GETP$:IFP$=""THEN1720
1730 RETURN
1740 PRINT"[illegible]EXISTIERT BEREITS EINE DATEI":PRINT"[illegible]AUF DER DISKETTE ? (J/N)
1750 GETX$:IFX$=""THEN1750
1760 IFX$="N"THENGOSUB1880
1770 IFX$<>"N"ANDX$<>"J"THEN1750
1780 OPEN1,8,1,"@0:DMM-DATEN,S,W"
1790 FORI=1TOA:PRINT#1,A(I):PRINTI,:NEXT
1800 FORI=1TOA:FORJ=1TOB:FORK=1TO3:PRINT#1,C(I,J,K):PRINTI,:NEXTK,J,I
1810 FORI=1TOA:FORJ=1TOB:PRINT#1,E(I,J):PRINTI,:NEXTJ,I
1820 CLOSE1:P0=1:RETURN
1830 OPEN1,8,0,"DMM-DATEN,S,R"
1840 FORI=1TOA:PRINTI,:INPUT#1,A(I):NEXT
1850 FORI=1TOA:FORJ=1TOB:FORK=1TO3:INPUT#1,C(I,J,K):PRINTI,:NEXTK,J,I
1860 FORI=1TOA:FORJ=1TOB:INPUT#1,E(I,J):PRINTI,:NEXTJ,I
1870 CLOSE1:P0=1:RETURN
1880 OPEN1,8,1,"DMM-DATEN,S,W":PRINT#1,"FUER DATEN RESERVIERT!"
1890 CLOSE1:RETURN
3000 OPEN1,4:PRINT#1:PRINT#1:PRINT#1,"STAND "LEFT$(TI$,2)" UHR "MID$(TI$,3,2)
3005 PRINT#1:GOSUB4000:FORI=FTOG-1:FORJ=I+1TOG:IFE$(J)<=E$(I)THEN3020
3010 RI$=E$(I):E$(I)=E$(J):E$(J)=RI$
3020 NEXTJ,I:PRINT#1,"AUSGEFUEHRTE DISZIPLINEN:"
3030 V=0:FORI=1TOB:FORJ=FTOG:IFE(J,I)<>0THENV1=V1+1
3040 NEXTJ:IFV1=G1THENPRINT#1,RIGHT$(B$(I),LEN(B$(I))-4),:V=V+1
3050 V1=0:NEXTI:PRINT#1
3060 PRINT#1:FORI=FTOG:PRINT#1,E$(I):NEXT
3070 IFV=1THENPRINT#1,"STAND NACH DER ERSTEN DISZIPLIN":GOTO3100
3080 IFV=0THENPRINT#1,"NOCH KEINE DISZIPLIN KOMPLETT!":GOTO3100
3090 PRINT#1,"STAND NACH"V" DISZIPLINEN!"
3100 CLOSE1:RETURN
4000 PRINT#1,"ZWISCHENERGEBNIS:":PRINT#1,CHR$(1)G$(X1):FORI=FTOG
4010 E1$="       "+STR$(A(I)):E$(I)=RIGHT$(E1$,7)+" "+A$(I):NEXT:PRINT#1:RETURN
READY.
```

Listing des DMM-Programms (Schluß)

100-140	Ausgabe der Copyrights auf dem Bildschirm, setzen der Hintergrundfarben und Eingabe der gegenwärtigen Zeit. Der WAIT-Befehl wartet auf einen Tastendruck und muß bei der Anpassung an andere Computer so wie die beiden POKE-Befehle entfernt werden
150-240	Festlegung der Mannschaften in DATAs, Bestimmung der Arrays durch die DIM-Anweisung und Einlesen der Mannschaften
250-390	Aufbau der Bildschirmmaske für das Hauptmenü
400-440	Auswahl des Unterprogramms, das angewählt wurde
450-510	Aufbau der Untermenübildschirmmaske und Verzweigung zu den jeweiligen Unterprogrammen, die von diesem angesprungen werden können
520-570	Auswahl der Disziplin und Ausgabe der Liste aller Disziplinen einer Gruppe
580-690	Eingabe der Ergebnisse, Sortieren der Ergebnisse und Ausgabe der sortierten Ergebnisse. Außerdem Addition zum Gesamtergebnis und Ausgabe des Gesamtergebnisses
700-840	Ausgabe des Zwischenergebnisses auf dem Bildschirm
850-860	Anschalten des Tones
870-940	Ausgabe der Fehlermeldung und des Tones im Falle einer Falscheingabe. Vorbereitung auf die Neueingabe der Ergebnisse
950-1020	Auslisten der Einzelergebnisse zur Kontrolle
1030-1040	Ändern des konkreten Ergebnisses
1050-1390	Unterprogramm zur Definition der Disziplinen und der Mannschaftsanzahl einer Gruppe
1510-1570	Schreiben der Daten auf Kassette
1580-1640	Lesen der Daten von Kassette
1680-1730	Unterprogramm zur Vorbereitung auf die Datenspeicheroperationen
1740-1820	Schreiben der Daten auf die Diskette in eine sequentielle Datei, die den Namen DMM-Daten trägt
1830-1870	Lesen der Daten von Diskette
1880-1890	Eröffnen einer Pseudo-Datei auf der Diskette, falls noch keine Datei angelegt war
3000-3100	Ausgabe der Zwischenergebnisse auf dem Drucker
4000-4010	Unterprogramm zum Drucken der Überschrift

Erläuterung des DMM-Programms anhand der Zeilennummern

sor rot gefärbt und die Eingabe wird ignoriert.

Die weiteren Programmschritte, die vom Hauptprogramm aus angewählt werden können, sind Einlesen von Daten, Abspeichern von Daten und Programmende.

Alle Peripheriegeräte können angesprochen werden

Die Datenspeicherung (Bild 10) kann sowohl auf Kassette als auch auf Diskette erfolgen. Die Möglichkeit einer solchen Speicherung trägt dazu bei, die Daten vor einem Stromausfall zu schützen und die Ergebnisse zu späteren Zeitpunkten wieder rekonstruieren zu können.

Um die Daten der Mannschaften zu ändern, muß zunächst die Variable A in Zeile 130 an die neue Gesamtanzahl der Mannschaften angepaßt werden. Die neuen Namen müssen in den DATA-Zeilen von 150 bis 220 geändert werden. Die dort vertretenen »..........« erklären sich dadurch, daß am Veranstaltungstag noch in letzter Minute bereits gemeldete Mannschaften ausfielen. Ihre Namen wurden dann einfach durch diese Striche ersetzt und ihre Ergebnisse wurden immer mit 0 eingegeben. Die Reihenfolge, in der die Mannschaften in den DATA-Zeilen stehen, entscheidet über die Reihenfolge der Mannschaftsnummern. Es ist hierbei darauf zu achten, daß Mannschaften, die in derselben Gruppe starten auch in der Reihenfolge der Nummern direkt nacheinander kommen. Die schließlich letzten Änderungen beziehen sich auf die Zeilen 1050, 1110, 1170, 1220 und 1350. In diesen Zeilen wird festgelegt, welche Mannschaften zu welcher Gruppe gehören und wieviele Disziplinen zu dieser Gruppe gehören. Die Anzahl der Disziplinen ist einfach dadurch zu ändern, daß man der Variable ***H*** einen anderen Wert zuweist. Die Namen selbst werden in den darauffolgenden Zeilen definiert, indem sie dem Feld der B$(x)-Variablen zugewiesen werden. Die höchstmögliche Anzahl an Disziplinen muß dann noch in Zeile 130 bei der Zuweisung zu der Variablen ***B*** geändert werden. In den oben genannten Zeilen wird z.B. in der Zeile 1050 durch die Zuweisung ***F=1*** und ***G=5*** festgelegt, daß die Mannschaften 1 bis 5 einschließlich zur Bundesliga Männer gehören. In der Zeile 1110 bedeutet ***F=6:G=10***, daß die Mannschaften 6 bis 10 zur Landesliga Männer gehören. Dies gilt ähnlich für die übrigen, oben genannten Zeilen. Die Reihenfolge der Gruppen ist hierbei die gleiche wie die im Hauptmenü verwendete.

(Volker Ritzhaupt)

A$(x)	Name der Vereine
A(x)	Gesamtergebnis eines Vereins
A	Anzahl der Mannschaften
B$(x)	Namen der Disziplinen einer Gruppe
B	Maximale Anzahl an Disziplinen
C(x,y,z)	Einzelergebnis, wobei x die Mannschaftsnummer ist y die Disziplin repräsentiert und z die Nummer des Einzelergebnisses ist (von 1-3)
C$	Variable für 4 Cursorbewegungen nach unten
E$(x)	Ergebniszeichenkette, die sich aus Gesamtpunktzahl und Mannschaftsnamen zusammensetzt
E(x,y)	Flag zur Überprüfung, ob eine Disziplin bereits eingegeben wurde x steht für die Mannschaftsnummer und y für die Disziplin
E1$	Zwischenspeicherstring zur Formatierung der Ergebnisse
F	Untere Grenze der Mannschaftsnummer einer Gruppe
G	Obere Grenze der Mannschaftsnummern einer Gruppe
F$	Variable für Bildschirm löschen und 3 Cursorbewegungen nach unten
G$(x)	Namen der unterschiedlichen Gruppen
G1	Anzahl der Mannschaften einer Gruppe
H	Anzahl der Disziplinen einer Gruppe
S1	Basisadresse des SID im C 64 (Sound Controller)
T	Wellenform des Tones
TI$	Interne Uhr des C 64
T1$	Stunden
T2$	Minuten
T3$	Sekunden
V	Anzahl der durchgeführten Disziplinen
V1	Zählvariable
X1	Angewähltes Unterprogramm beziehungsweise angewählte Gruppe
P0,K0	Flag für bestimmten Programmodus
X$,Y$,D$,P$,RI$	Stringvarible ohne bleibenden Wert. (Ja/Nein-Entscheidungen etc.)
I,X,D,J,RI	Numerische Variable ohne bleibenden Wert (Laufvariable etc.)

Variablendefinition zum DMM-Programm

BASIC ohne Probleme!



Ch. Langfelder

Band 1: Unterweisung

Eine Einführung in BASIC mit CBM-Rechnern (CBM 8032)

In 12 Kapiteln wird der Leser Schritt für Schritt mit der Programmiersprache BASIC, dem CBM-Rechner und seiner Bedienung vertraut gemacht. Jedes Kapitel schließt mit Übungen und Aufgaben ab — als Kontrolle für den jeweiligen Wissensstand. Im Anhang befinden sich dann unter anderem die Lösungen der Aufgaben, ein Glossar, ein Stichwortregister usw.

DM 36,—*

1983, 226 Seiten
Best.-Nr. MT 480



Ch. Langfelder

Band 2: Übungen

Dieses Buch enthält 20 ausgewählte Routinen und Programme zum Üben allgemeiner Programmiertechniken auf CBM-Rechnern (CBM 8032). Die Programme sind in sechs Rubriken unterteilt: drei allgemeine Routinen, fünf allgemeine Programme, fünf kommerziell-technische Anwendungen, zwei Statistikprogramme, zwei Mathematikprogramme und drei Lehr- und Spielprogramme. Alle Programme können direkt in einen CBM-Rechner Modell 8032 eingegeben und gestartet werden.

DM 26,—*

1982, 119 Seiten
Best.-Nr. MT 490





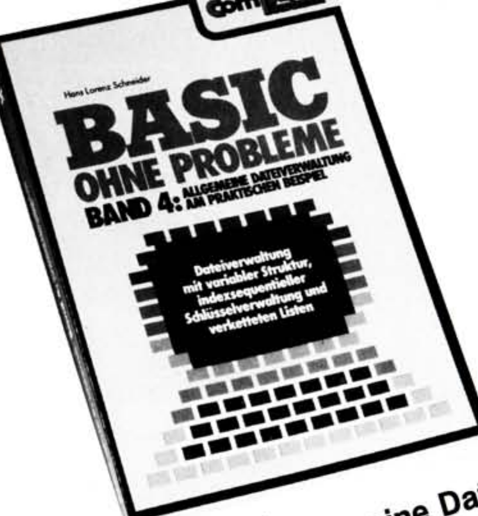
H.L. Schneider

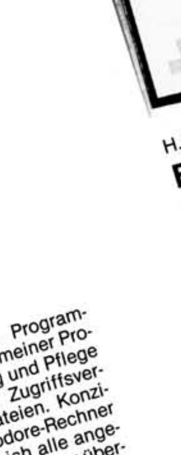
Band 3: Programmentwicklung und Datenverwaltung

Aus dem Inhalt: Problemanalyse · Programmiertechniken · Beschreibung allgemeiner Programmsegmente · Testen, Wartung und Pflege von Programmen · Dateiformen · Zugriffsverfahren auf eine oder mehrere Dateien. Konzipiert ist dieses Buch für Commodore-Rechner der 8000er-Serie, jedoch lassen sich alle angegebenen Algorithmen auf andere Rechner übertragen.

DM 44,—*

1983, 256 Seiten
Best.-Nr. MT 500





H.L. Schneider

Band 4: Allgemeine Dateiverwaltung am praktischen Beispiel

Das vorliegende Buch faßt die meisten der in »Basic ohne Probleme« Band 3 beschriebenen Algorithmen in ein großes, komplexes Programmsystem zusammen, das immer wieder und in allen Bereichen der Datenverarbeitung benötigt wird: die Dateiverwaltung. Alle Programme werden ausführlich beschrieben. Im Gegensatz zu Standard-Dateiverwaltungen können Sie aufgrund der ausführlichen Dokumentation Ihre Dateiverwaltung immer Ihren Wünschen anpassen.

DM 53,—*

1983, 428 Seiten
Best.-Nr. MT 514

64'er
Magazin für Computerfans

Hans-Pinsel-Straße 2
8013 Haar bei München
Telefon 089/4613-220

* alle Preise inkl. MwSt. zuzügl. Versandspesen



Schreiben Sie ein tolles neues Video- oder Computerspielprogramm. Damit werden Sie dann vielleicht zum Millionär. Dieser Super-Wettbewerb, der von IRP (The International Register of independent Computer Programmers Ltd.) und der bekannten internationalen Mark McCormack Management Group organisiert wird, bietet Programmierern und Leuten mit Ideen die Chance ihres Lebens. Zu gewinnen sind sowohl riesige Geldpreise als auch Absatzbeteiligungen von 10 % am weltweiten Verkauf der Spiele. Als Gewinner werden Sie auch zum Star einer internationalen Fernsehshow. Werden Sie reich und berühmt mit Ihrem Können und Ihrer Phantasie!

Hauptgewinn: 100 000 Dollar

PLUS und dann gibt es noch fünf weitere Preise zu je 15 000 Dollar!

Denken Sie sich ein völlig neues, originelles Video- oder Computerspiel aus: Es kann ein Sport-, Simulator-, Arkaden-, Strategie-, Abenteuer- oder Fantasyspiel sein. Es muß auch nicht unbedingt ein Spielprogramm sein, sondern es könnte auch ein Lernspiel oder Lernprogramm sein, aber sehr unterhaltsam sollte es sein. Wir vergeben auch eine Reihe von Auszeichnungen für besonders gelungene Spielprogramme, die vom Händler mit einem speziellen Hinweis (»An international Video Game of the Year MERIT AWARD«) auf der Packung gekennzeichnet werden dürfen. Das ist eine große Herausforderung. Nicht nur die riesigen Geldgewinne sind attraktiv, sondern Sie können über Nacht berühmt werden. Das ist der aufregendste Wettbewerb, der je für kreative Video- und Computerspielefans ausgeschrieben wurde.

Werden Sie auch noch zum Fernsehstar!

Die sechs preisgekrönten Spiele und ihre Autoren werden in einer international ausgestrahlten Fernsehshow vorgestellt. Dann werden Sie plötzlich einen Namen haben, den man nicht mehr vergißt.

Teilnahmebedingungen:

Schicken Sie einfach Ihr Spiel auf Kassette — egal für welchen Heimcomputer — ein. Verwenden Sie diesen Abschnitt heute noch, und wir schicken alle weiteren Einzelheiten.

Einsendeschluß: 31. Mai 1984

an: IRP Limited,
Pinewood Film Studios,
Iver, Bucks, England

Name: ....

Adresse: ....

....

Ein schneller »Drawline«-Algorithmus

Im folgenden wird eine Möglichkeit vorgestellt, schnell und einfach eine Strecke, die durch ihre beiden Endpunkte gegeben ist, zu plotten. Als Ausgabegerät können Bildschirm, Drucker oder Plotter eingesetzt werden.

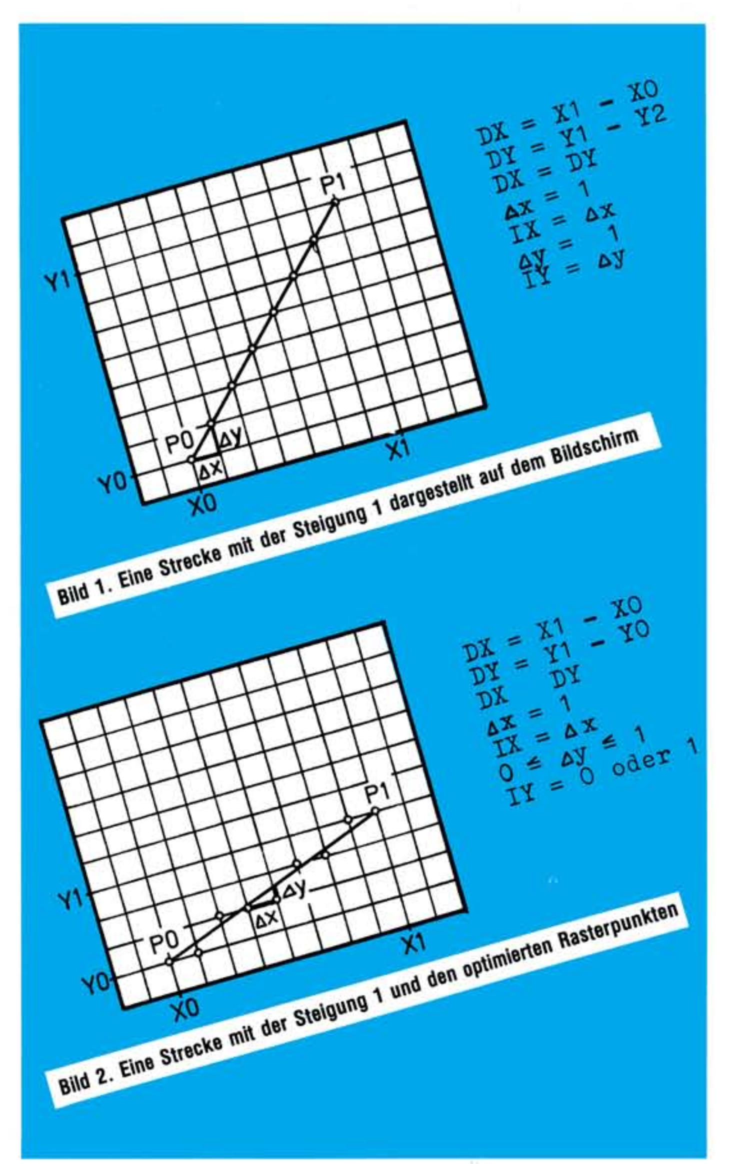


Bild 1. Eine Strecke mit der Steigung 1 dargestellt auf dem Bildschirm

Bild 2. Eine Strecke mit der Steigung 1 und den optimierten Rasterpunkten

Der Algorithmus wird in einer Basic- und einer Assembler-Version für einen 6502-Mikroprozessor mit den Adressen für den Commodore 64 (der 6510 Mikroprozessor im C 64 ist im Befehlssatz identisch mit dem 6502) beschrieben. Die Programme können, da sie im Aufbau einfach sind und erklärt werden, ohne große Probleme für andere Systeme beziehungsweise andere Sprachen umgewandelt werden.

Vor einiger Zeit suchte ich eine Möglichkeit, das in [1] auf Seite 97 abgedruckte, in Assembler geschriebene Programm um eine einfache aber doch effiziente »Drawline«-Routine zu erweitern. In [2] fand ich genau den Algorithmus, den ich brauchte, — nur war er »leider« in Basic formuliert. Jedoch bereitete es, nachdem der Algorithmus verstanden war, nicht mehr allzu viel Mühe, eine Assemblerversion zu schreiben. Doch vor der Beschreibung der Programme eine kurze Erklärung des Algorithmus.

Ein Rasterbildschirm setzt sich aus einzelnen Punkten, die gesetzt oder auch gelöscht sein können, zusammen. Der Abstand der Punkte voneinander ist in der Richtung der Achsen immer gleich und jeweils eine Schrittweite groß (Bild 1). Um eine Gerade zwischen den Punkten P0 und P1 zu ziehen, muß daher schrittweise berechnet werden, welcher Punkt der Ideallinie (Bild 2) am nächsten ist und daher gesetzt werden muß.

Geschwindigkeitsvorteile durch einfache Berechnungen

Für diese Berechnungen gibt es verschiedene Möglichkeiten, doch sind sie meistens mit Multiplikationen und Divisionen in der Approximationsschleife verbunden und daher weder schnell noch einfach zu programmieren. Der hier vorgestellte Algorithmus verwendet dagegen nur eine Division und in der Schleife nur mehr Addition, Subtraktion und eine Vergleichsoperation. Da die Schleifenoperationen außerdem nur mehr an Integerzahlen durchzuführen sind, ist er besonders schnell, und er läßt sich auch einfach programmieren. Zur Erklärung soll eine Gerade mit einer Steigung zwischen 0 und 1 (0 bis 45 Grad) betrachtet werden.

Wie man in Bild 2 unschwer erkennen kann, ist der Abstand der Punkte P0 und P1 entlang der X-Achse gleich der Anzahl der zu setzenden Punkte, DX = X1 — X0, das heißt es sind DX-Approximationen durchzuführen, um die Gerade zu zeichnen. Für jeden folgenden Punkt ist also X0 um IX (= 1) zu erhöhen, während Y0 gleichbleibt (IY = 0) oder ebenfalls um 1 (IY = 1) erhöht wird. Der Abstand der beiden Punkte entlang der Y-Achse ist demnach: DY = Y1 — Y0. Es bleibt also nur mehr festzustellen, wann IY = 0, beziehungsweise IY = 1 zu sein hat. Dazu wird vor Beginn der Schleife ein Approximationswert OF berechnet. Da beim Idealfall für eine Steigung von 1 (Bild 1) IY = IX, das heißt immer 1 ist, und DX = DY ist, wird OF = DX/2.

Rasterpunkte optimieren

Zu OF wird für jeden neuen Punkt DY addiert. Solange OF kleiner als DX bleibt, bleibt IY = 0, wird OF gleich oder größer, so wird IY = 1, das heißt Y0 um 1 erhöht. Damit diese Abfrage auch für die folgenden Punkte möglich ist, muß OF um DX vermindert werden.

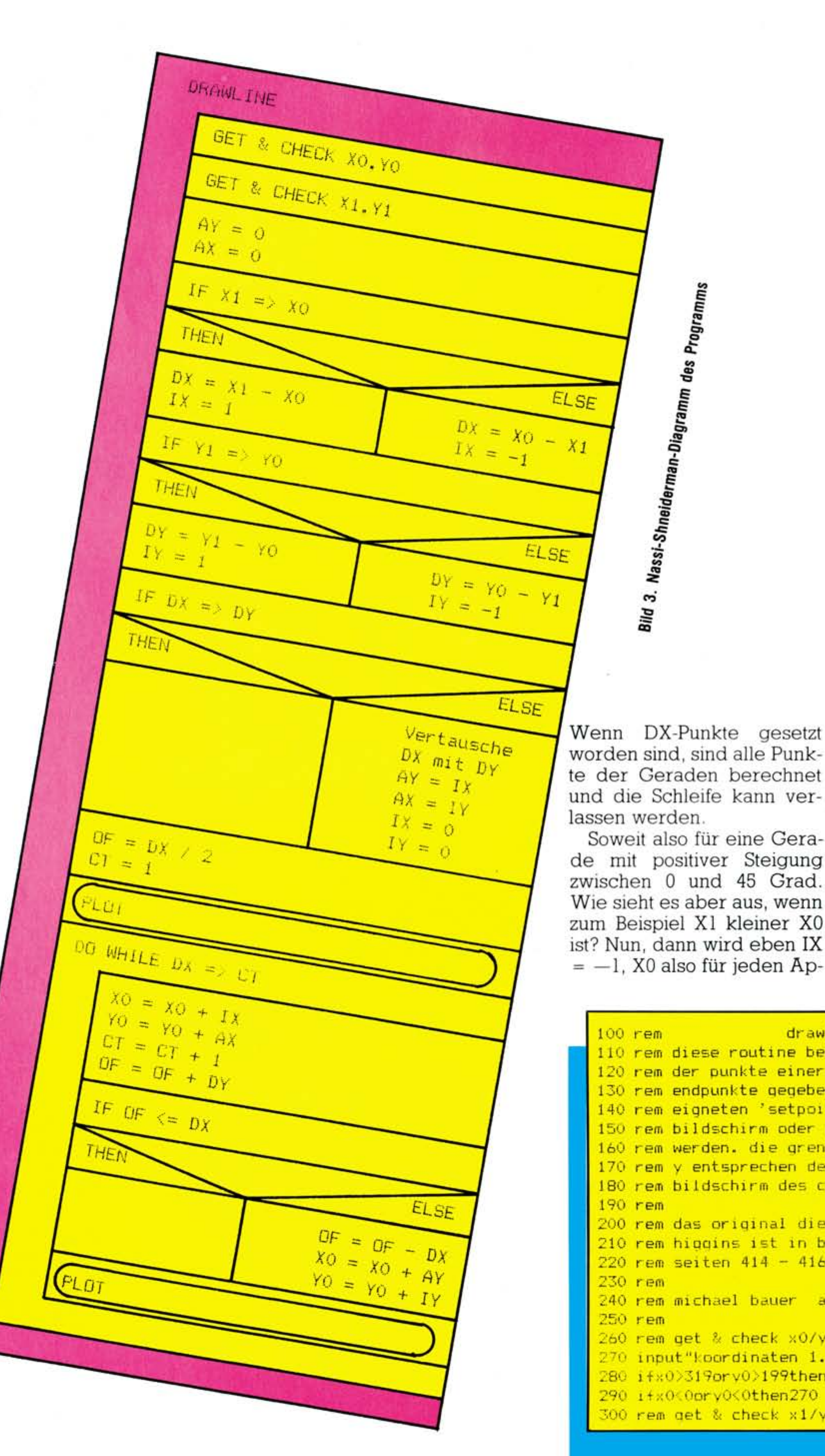


Bild 3. Nassi-Shneiderman-Diagramm des Programms

Wenn DX-Punkte gesetzt worden sind, sind alle Punkte der Geraden berechnet und die Schleife kann verlassen werden.

Soweit also für eine Gerade mit positiver Steigung zwischen 0 und 45 Grad. Wie sieht es aber aus, wenn zum Beispiel X1 kleiner X0 ist? Nun, dann wird eben IX = —1, X0 also für jeden Approximationsschritt um 1 erniedrigt. Und wie sieht es für eine Steigung größer 1 aus? Auch dieses Problem läßt sich einfach lösen. Es werden für die Rechnung einfach die beiden Achsen vertauscht.

Der Ablauf beider Programme ist im Nassi-Shneiderman-Diagramm (Bild 3) dargestellt. Zum leichteren Vergleich sind alle Variablenbezeichnungen in den Programmen identisch.

Die Basic-Version gibt nur die Koordinaten der berechneten Punkte aus, da ich dafür keine »Setze-Punkt«-Routine schreiben wollte. Man kann aber die Wirkungsweise des Algorithmus schön verfolgen.

Das Assemblerprogramm ist, wie schon oben gesagt, für den C 64 geschrieben mit einer möglichen Auflösung von 320 x 200 Punkten. Dadurch können die Werte für Y in einem Byte untergebracht werden, während für X zwei Byte benötigt werden. OF, DX und der Schleifenzähler CT benötigen deshalb ebenfalls 2 Byte. Die Länge der Werte muß beim Umstricken für ein anderes System berücksichtigt werden, da alle durchzuführenden Operationen dementsprechend 1 oder 2 Byte lang sind.

```
100 rem              drawline
110 rem diese routine berechnet die koordinaten
120 rem der punkte einer strecke, die durch ihre
130 rem endpunkte gegeben ist. mit einer ge-
140 rem eigneten 'setpoint'-routine kann der
150 rem bildschirm oder ein plotter angesteuert
160 rem werden. die grenzen der werte fuer x und
170 rem y entsprechen den werten fuer den hi-res
180 rem bildschirm des commodore 64.
190 rem
200 rem das original dieses programmes von mike
210 rem higgins ist in byte heft 8/81 auf den
220 rem seiten 414 - 416 erschienen.
230 rem
240 rem michael bauer  aindorferstr. 86  8 muenchen 21
250 rem
260 rem get & check x0/y0
270 input"koordinaten 1. punkt";x0,y0
280 ifx0>319ory0>199then270
290 ifx0<0ory0<0then270
300 rem get & check x1/y1
```

Name	Adressen 2001	3032	8032	VC20	C 64	610/710	Beschreibung
CHKCOM	CE11	CDF8	BEF5	CEFD	AEFD	9730	Pruefe ob naechstes Zeichen im BASIC-Text ein Komma ist, wenn nicht gebe 'SYNTAX ERROR' aus
GETCOR	D6C4	D6C6	C921	D7EB	B7EB	B4E5	Holt die Koordinaten eines Punktes aus dem BASIC-Text. Die Routine wertet auch Ausdruecke aus. Die X-Koordinate wird als 2-Byte-Wert in X0 , die Y-Koordinate als Byte im X-Register uebergeben.
X0	0066	0014	0014	0014	0014	0011	Hier wird die X-Koordinate von GETCOR abgelegt.
PLOT	--	--	--	--	--	--	Diese Routine ist keine Betriebssystemroutine. Sie uebernimmt die Koordinaten von X0 und dem X-Register.

Tabelle 1. Benutzte Unterroutinen und Übergabespeicherzellen

Name	Beschreibung
X0,Y0	Koordinaten des ersten Punktes
X1,Y1	Koordinaten des zweiten Punktes
CT	Schleifenzaehler
IX	Inkrement oder Dekrement fuer X0 (-1,0,+1)
IY	Inkrement oder Dekrement fuer Y0 (-1,0,+1)
AX	wie IX fuer Steigungen > 1
AY	wie IY fuer Steigungen > 1
DX	Entfernung der Punkte entlang der X-Achse (= Anzahl der Punkte)
DY	Entfernung der Punkte entlang der Y-Achse
OF	Approximationsvariable zur Bestimmung ob Y0 gleichbleibt

Tabelle 2. Die verwendeten Variablen

```
310 input"koordinaten 2. punkt";x1,y1
320 ifx1>319ory1>199then310
330 ifx1<0ory1<0then310
340 rem initialisiere variable
350 ay=0:iy=1:ix=1:ax=0
360 rem pruefe steigung
370 ifx1=>x0thendx=x1-x0:goto400
380 ix=-1
390 dx=x0-x1
400 ify1=>y0thendy=y1-y0:goto440
410 dy=y0-y1
420 iy=-1
430 rem steigung > 1 ?
440 ifdx=>dythen530
450 ct=dx:rem vertausche dx und dy
460 dx=dy
470 dy=ct
480 ay=ix
490 ix=0
500 ax=iy
510 iy=0
520 rem berechne approximationswert
530 of=dx/2
540 ct=1:rem schleifenzaehler
550 goto660:rem plotte ursprungspunkt
560 rem ******** approximationsschleife
570 x0=x0+ix
580 y0=y0+ax
590 of=of+dy
600 ct=ct+1
610 rem y0 erhoehen ?
620 ifof<=dxthen660
630 of=of-dx
640 x0=x0+ay
650 y0=y0+iy
660 printx0,y0
670 rem letzter punkt ?
680 ifdx=>ctthen570
690 end
```

Basic-Programm »Drawline«

In dieser Assemblerversion werden nur 2 Subroutinen aus dem Betriebssystem verwendet. Sie sind in Tabelle 1 beschrieben. Zusätzlich habe ich die Adressen für die anderen Commodore Computer angegeben. Die Subroutine »PLOT« muß, wenn sie sich im Betriebssystem wie bei den Commodoresystemen nicht findet, extra geschrieben werden. Für den C 64 findet man in [1] ein geeignetes Programm.

Kein Problem: Variablen

In Tabelle 2 sind alle verwendeten Variablen aufgelistet und beschrieben. Aufgerufen wird diese Version mit
SYS aaaa,X0,Y0,X1,Y1 wobei aaaa die Startadresse der Routine, X0/Y0 und X1/Y1 die Koordinaten der beiden Punkte sind. Für die Koordinaten können auch Ausdrücke verwendet werden, da die Betriebssystemroutine »GETCOR« auch Ausdrücke auswertet.

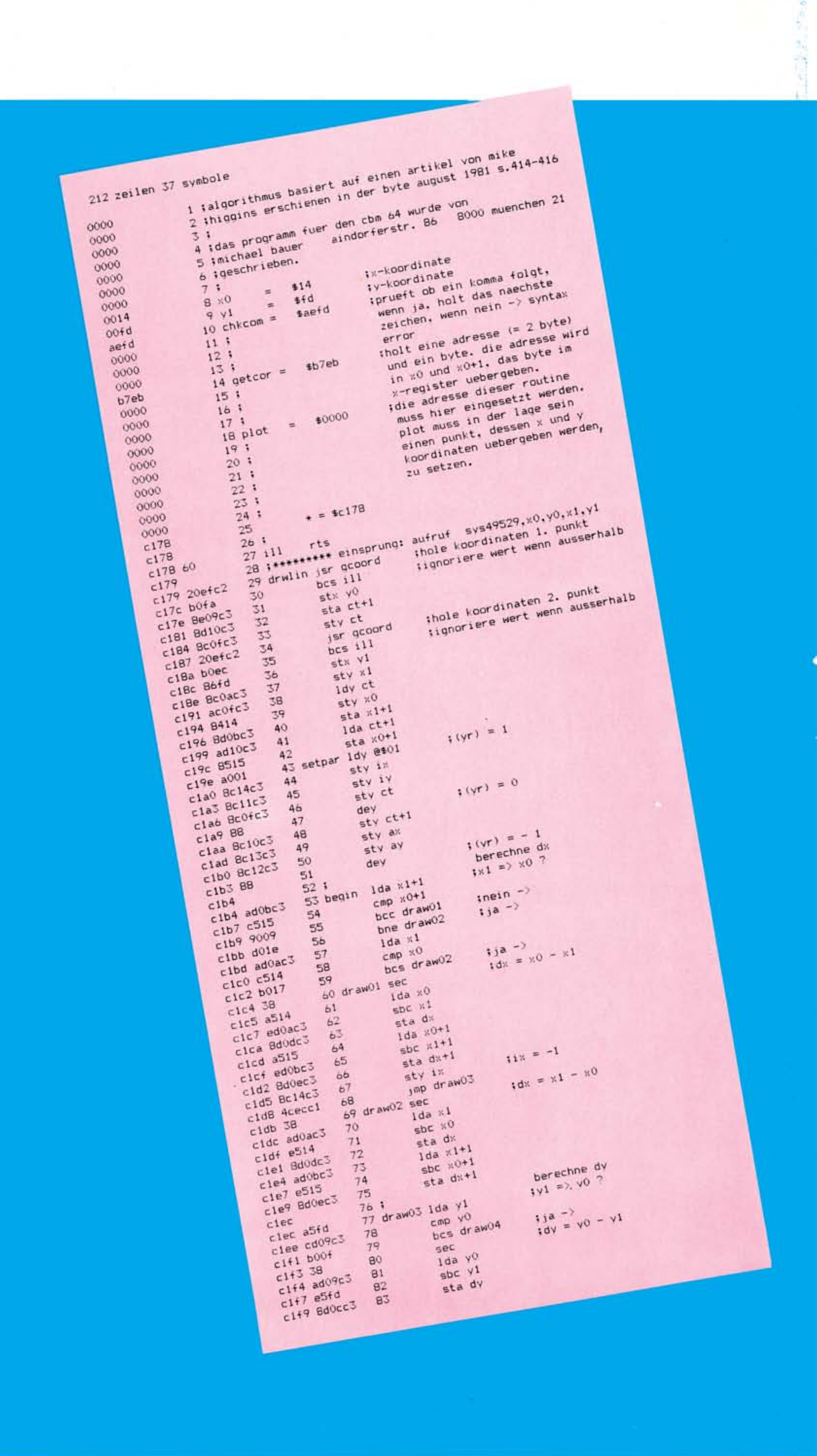
Zusätzliche Erweiterungen

Den Lesern, die mit Assembler-Programmen noch etwas Probleme haben, sende ich gerne gegen einen Kostenbeitrag von 10 Mark eine Kassette mit einem Basic-Lader mit dem gesamten Programm zu. Es stehen dann neben den in [1] beschriebenen Befehlen noch DRAWLINE, ERASELINE, DRAWX-AXIS, ERASEX-AXIS, DRAWY-AXIS und ERASEY-AXIS zur Verfügung.

Bei der Anwendung dieses Algorithmus wünsche ich viel Spaß und Erfolg.

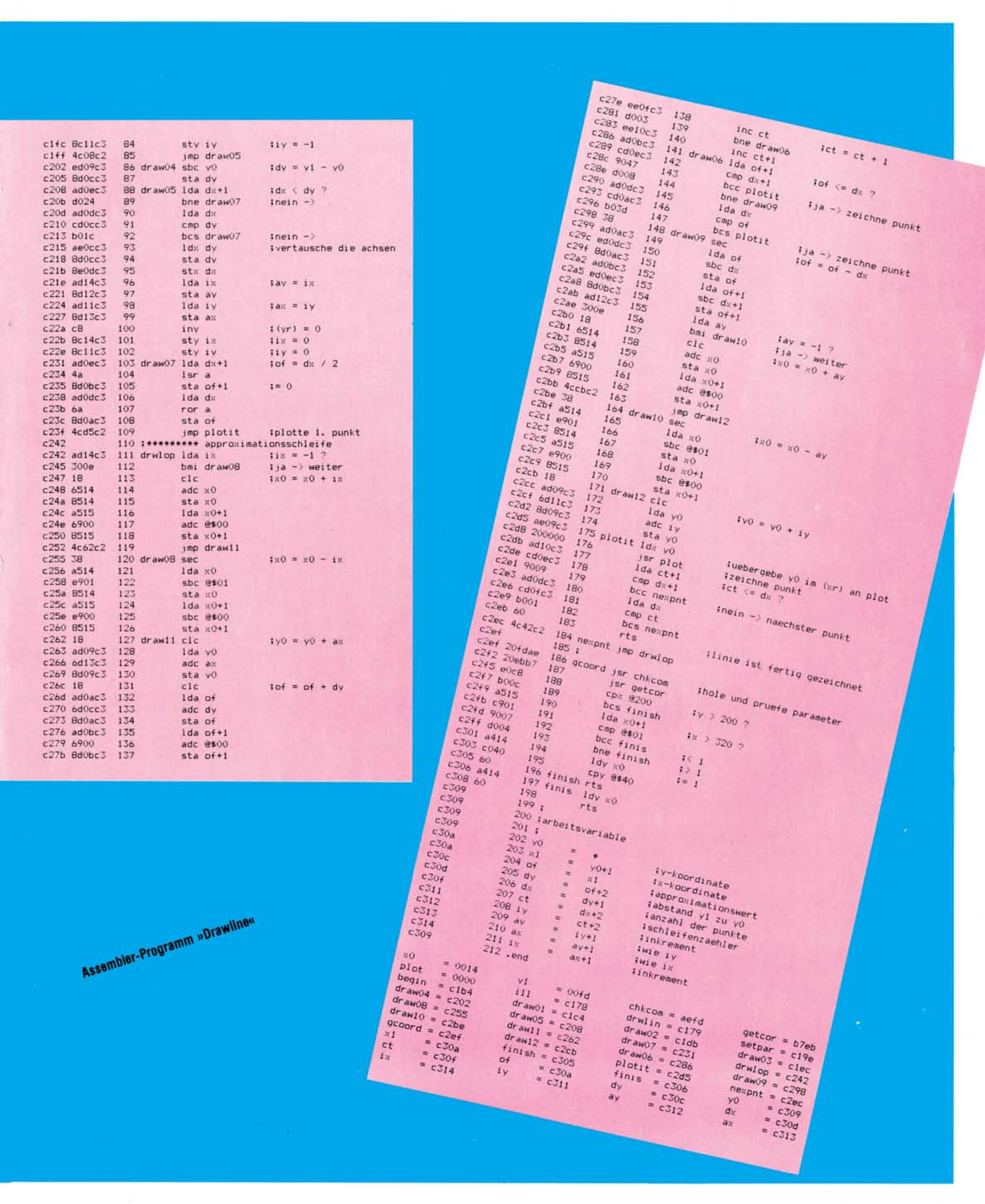
(Michael Bauer)

Literatur:
(1) Angerhausen et al.; »64 Intern« Seiten 97-100; DATA BECKER 1983
(2) Higgins, Mike; »Fast Line-Drawing Technique« Seiten 414-416; BYTE August 1981



```
212 zeilen 37 symbole

0000            ;algorithmus basiert auf einen artikel von mike
0000            ;higgins erschienen in der byte august 1981 s.414-416
0000            ;
0000            ;das programm fuer den cbm 64 wurde von
0000            ;michael bauer     aindorferstr. 86   8000 muenchen 21
0000            ;geschrieben.
0000            ;                          ;x-koordinate
0014            x0     =   $14             ;y-koordinate
00fd            y1     =   $fd             ;prueft ob ein komma folgt,
aefd            chkcom =   $aefd            wenn ja, holt das naechste
0000            ;                           zeichen, wenn nein -> syntax
0000            ;                           error
0000            ;                          ;holt eine adresse (= 2 byte)
b7eb            getcor =   $b7eb            und ein byte. die adresse wird
0000            ;                           in x0 und x0+1. das byte im
0000            ;                           x-register uebergeben.
0000            ;                          ;die adresse dieser routine
0000            plot   =   $0000            muss hier eingesetzt werden.
0000            ;                           plot muss in der lage sein
0000            ;                           einen punkt, dessen x und y
0000            ;                           koordinaten uebergeben werden,
0000            ;                           zu setzen.
0000            ;
0000            ;         * = $c178
0000
c178            ;
c178            ill    rts
c178 60         ;********* einsprung: aufruf  sys49529,x0,y0,x1,y1
c179            drwlin jsr qcoord          ;hole koordinaten 1. punkt
c179 20efc2            bcs ill             ;ignoriere wert wenn ausserhalb
c17c b0fa              stx y0
c17e 8e09c3            sta ct+1
c181 8d10c3            sty ct
c184 8c0fc3            jsr qcoord          ;hole koordinaten 2. punkt
c187 20efc2            bcs ill             ;ignoriere wert wenn ausserhalb
c18a b0ec              stx y1
c18c 86fd              sty x1
c18e 8c0ac3            ldy ct
c191 ac0fc3            sty x0
c194 8414              sta x1+1
c196 8d0bc3            lda ct+1
c199 ad10c3            sta x0+1
c19c 8515       setpar ldy @$01            ;(yr) = 1
c19e a001              sty ix
c1a0 8c14c3            sty iy
c1a3 8c11c3            sty ct
c1a6 8c0fc3            dey                 ;(yr) = 0
c1a9 88                sty ct+1
c1aa 8c10c3            sty ax
c1ad 8c13c3            sty ay
c1b0 8c12c3            dey                 ;(yr) = - 1
c1b3 88         ;                           berechne dx
c1b4            begin  lda x1+1            ;x1 => x0 ?
c1b4 ad0bc3            cmp x0+1
c1b7 c515              bcc draw01          ;nein ->
c1b9 9009              bne draw02          ;ja ->
c1bb d01e              lda x1
c1bd ad0ac3            cmp x0
c1c0 c514              bcs draw02          ;ja ->
c1c2 b017       draw01 sec                 ;dx = x0 - x1
c1c4 38                lda x0
c1c5 a514              sbc x1
c1c7 ed0ac3            sta dx
c1ca 8d0dc3            lda x0+1
c1cd a515              sbc x1+1
c1cf ed0bc3            sta dx+1
c1d2 8d0ec3            sty ix              ;ix = -1
c1d5 8c14c3            jmp draw03
c1d8 4cecc1     draw02 sec                 ;dx = x1 - x0
c1db 38                lda x1
c1dc ad0ac3            sbc x0
c1df e514              sta dx
c1e1 8d0dc3            lda x1+1
c1e4 ad0bc3            sbc x0+1
c1e7 e515              sta dx+1
c1e9 8d0ec3     ;                           berechne dy
c1ec            draw03 lda y1              ;y1 =>, y0 ?
c1ec a5fd              cmp y0
c1ee cd09c3            bcs draw04          ;ja ->
c1f1 b00f              sec                 ;dy = y0 - y1
c1f3 38                lda y0
c1f4 ad09c3            sbc y1
c1f7 e5fd              sta dy
c1f9 8d0cc3
```



```
c1fc 8c11c3   84        sty iy         ;iy = -1
c1ff 4c08c2   85        jmp draw05
c202 ed09c3   86 draw04 sbc y0         ;dy = y1 - y0
c205 8d0cc3   87        sta dy
c208 ad0ec3   88 draw05 lda dx+1       ;dx < dy ?
c20b d024     89        bne draw07     ;nein ->
c20d ad0dc3   90        lda dx
c210 cd0cc3   91        cmp dy
c213 b01c     92        bcs draw07     ;nein ->
c215 ae0cc3   93        ldx dy         ;vertausche die achsen
c218 8d0cc3   94        sta dy
c21b 8e0dc3   95        stx dx
c21e ad14c3   96        lda ix         ;ay = ix
c221 8d12c3   97        sta ay
c224 ad11c3   98        lda iy         ;ax = iy
c227 8d13c3   99        sta ax
c22a c8      100        iny            ;(yr) = 0
c22b 8c14c3  101        sty ix         ;ix = 0
c22e 8c11c3  102        sty iy         ;iy = 0
c231 ad0ec3  103 draw07 lda dx+1       ;of = dx / 2
c234 4a      104        lsr a
c235 8d0bc3  105        sta of+1       ;= 0
c238 ad0dc3  106        lda dx
c23b 6a      107        ror a
c23c 8d0ac3  108        sta of
c23f 4cd5c2  109        jmp plotit     ;plotte 1. punkt
c242         110 ;********* approximationsschleife
c242 ad14c3  111 drwlop lda ix         ;ix = -1 ?
c245 300e    112        bmi draw08     ;ja -> weiter
c247 18      113        clc            ;x0 = x0 + ix
c248 6514    114        adc x0
c24a 8514    115        sta x0
c24c a515    116        lda x0+1
c24e 6900    117        adc @$00
c250 8515    118        sta x0+1
c252 4c62c2  119        jmp draw11
c255 38      120 draw08 sec            ;x0 = x0 - ix
c256 a514    121        lda x0
c258 e901    122        sbc @$01
c25a 8514    123        sta x0
c25c a515    124        lda x0+1
c25e e900    125        sbc @$00
c260 8515    126        sta x0+1
c262 18      127 draw11 clc            ;y0 = y0 + ax
c263 ad09c3  128        lda y0
c266 6d13c3  129        adc ax
c269 8d09c3  130        sta y0
c26c 18      131        clc            ;of = of + dy
c26d ad0ac3  132        lda of
c270 6d0cc3  133        adc dy
c273 8d0ac3  134        sta of
c276 ad0bc3  135        lda of+1
c279 6900    136        adc @$00
c27b 8d0bc3  137        sta of+1
```

```
c27e ee0fc3  138        inc ct         ;ct = ct + 1
c281 d003    139        bne draw06
c283 ee10c3  140        inc ct+1
c286 ad0bc3  141 draw06 lda of+1       ;of <= dx ?
c289 cd0ec3  142        cmp dx+1
c28c 9047    143        bcc plotit     ;ja -> zeichne punkt
c28e d008    144        bne draw09
c290 ad0dc3  145        lda dx
c293 cd0ac3  146        cmp of
c296 b03d    147        bcs plotit     ;ja -> zeichne punkt
c298 38      148 draw09 sec            ;of = of - dx
c299 ad0ac3  149        lda of
c29c ed0dc3  150        sbc dx
c29f 8d0ac3  151        sta of
c2a2 ad0bc3  152        lda of+1
c2a5 ed0ec3  153        sbc dx+1
c2a8 8d0bc3  154        sta of+1
c2ab ad12c3  155        lda ay         ;ay = -1 ?
c2ae 300e    156        bmi draw10     ;ja -> weiter
c2b0 18      157        clc            ;x0 = x0 + ay
c2b1 6514    158        adc x0
c2b3 8514    159        sta x0
c2b5 a515    160        lda x0+1
c2b7 6900    161        adc @$00
c2b9 8515    162        sta x0+1
c2bb 4ccbc2  163        jmp draw12
c2be 38      164 draw10 sec            ;x0 = x0 - ay
c2bf a514    165        lda x0
c2c1 e901    166        sbc @$01
c2c3 8514    167        sta x0
c2c5 a515    168        lda x0+1
c2c7 e900    169        sbc @$00
c2c9 8515    170        sta x0+1
c2cb 18      171 draw12 clc            ;y0 = y0 + iy
c2cc ad09c3  172        lda y0
c2cf 6d11c3  173        adc iy
c2d2 8d09c3  174        sta y0
c2d5 ae09c3  175 plotit ldx y0         ;uebergebe y0 im (xr) an plot
c2d8 200000  176        jsr plot       ;zeichne punkt
c2db ad10c3  177        lda ct+1       ;ct <= dx ?
c2de cd0ec3  178        cmp dx+1
c2e1 9009    179        bcc nexpnt     ;nein -> naechster punkt
c2e3 ad0dc3  180        lda dx
c2e6 cd0fc3  181        cmp ct
c2e9 b001    182        bcs nexpnt
c2eb 60      183        rts            ;linie ist fertig gezeichnet
c2ec 4c42c2  184 nexpnt jmp drwlop
c2ef         185 ;
c2ef 20fdae  186 gcoord jsr chkcom     ;hole und pruefe parameter
c2f2 20ebb7  187        jsr getcor
c2f5 e0c8    188        cpx @200       ;y > 200 ?
c2f7 b00c    189        bcs finish
c2f9 a515    190        lda x0+1
c2fb c901    191        cmp @$01       ;x > 320 ?
c2fd 9007    192        bcc finis      ;< 1
c2ff d004    193        bne finish     ;> 1
c301 a414    194        ldy x0
c303 c040    195        cpy @$40       ;= 1
c305 60      196 finish rts
c306 a414    197 finis  ldy x0
c308 60      198        rts
c309         199 ;
c309         200 ;arbeitsvariable
c309         201 ;
c309         202 y0     =  *           ;y-koordinate
c30a         203 x1     =  y0+1        ;x-koordinate
c30a         204 of     =  x1          ;approximationswert
c30c         205 dy     =  of+2        ;abstand y1 zu y0
c30d         206 dx     =  dy+1        ;anzahl der punkte
c30f         207 ct     =  dx+2        ;schleifenzaehler
c311         208 iy     =  ct+2        ;inkrement
c312         209 ay     =  iy+1        ;wie iy
c313         210 ax     =  ay+1        ;wie ix
c314         211 ix     =  ax+1        ;inkrement
c309         212 .end

x0     = 0014   y1     = 00fd   chkcom = aefd   getcor = b7eb
plot   = 0000   ill    = c178   drwlin = c179   setpar = c19e
begin  = c1b4   draw01 = c1c4   draw02 = c1db   draw03 = c1ec
draw04 = c202   draw05 = c208   draw07 = c231   drwlop = c242
draw08 = c255   draw11 = c262   draw06 = c286   draw09 = c298
draw10 = c2be   draw12 = c2cb   plotit = c2d5   nexpnt = c2ec
gcoord = c2ef   finish = c305   finis  = c306   y0     = c309
x1     = c30a   of     = c30a   dy     = c30c   dx     = c30d
ct     = c30f   iy     = c311   ay     = c312   ax     = c313
ix     = c314
```

Assembler-Programm »Drawline«



Bild 1. In diesem Bildschirmbereich lassen sich die Sprites bewegen

Bild 2. Die Kennzahlen für die acht möglichen Richtungen

Sprites schneller bewegen

Daß Sprites nützliche Hilfsmittel nicht nur bei Spielen sind, ist mittlerweile bekannt. Mit dem hier vorgestellten Programm lassen sich bis zu acht Sprites mit 255 verschiedenen Geschwindigkeiten kontinuierlich in acht Richtungen manövrieren. Das Besondere: Berechnungen im Programm haben keinen Einfluß auf die Laufgeschwindigkeit der Sprites.

Eine feine Sache ist beim Commodore 64 die Sprite-Grafik. Hat man ein solches Sprite entwickelt und auf dem Bildschirm dargestellt, so kann durch Veränderung zweier Speicherstellen das Sprite an eine andere Position gebracht werden. Um eine fließende Bewegung zu erreichen, ändert man die Speicherstellen um den Faktor 1. Nun funktioniert dies bei einem einzigen Sprite noch recht zügig. Möchte man aber mehrere Sprites bewegen, zwischendurch etwas berechnen und abfragen, ob eine Kollision zwischen zwei Sprites stattgefunden hat, so werden die Bewegungen doch recht träge. Um die Bewegungen schneller erscheinen zu lassen, kann man die Schrittweite erhöhen. Dies hat aber den Nachteil, daß die Sprites ziemlich »abgehackt« über den Bildschirm laufen. Weiterhin muß beachtet werden, ob ein Sprite auf der rechten Bildschirmseite dargestellt werden soll. Ist dies der Fall, so muß im Register 16 das Bit für das entsprechende Sprite gesetzt werden. Bei der Rückkehr auf die linke Seite muß das Bit wieder gelöscht werden. Bild 1 zeigt den Bereich, in dem sich die Sprites bewegen können. Wenn Sie das Programm SPRITE-MOVE (siehe Listing) benutzen, so sind diese Probleme vorbei. Sie können ein Sprite nun mit dem Befehl

!RUN S,R,G

auf die Reise schicken. Mit dem S geben Sie die Nummer des Sprites an (0-7), mit R die Richtung (1-9) und mit G die Geschwindigkeit (0-255). In Bild 2 können Sie sehen, welche Zahl für welche Richtung einzusetzen ist. Die

```
5 REMSPRITE-MOVE
10 POKE183,11:POKE185,0:POKE186,1
20 POKE187,PEEK(43)+5:POKE188,PEEK(44)
30 POKE193,77:POKE194,174:POKE174,59
40 POKE175,207:SYS62954
50 REM ZEILE 5 BITTE MIT EINGEBEN
60 REM LADEN ERFOLGT MIT LOAD"SPRITE-MOVE",1,1
70 REM NACH DEM LADEN NEW EINGEBEN
```

Bild 3. Mit diesem kurzen Programm lassen sich die Sprites abspeichern

```
200 PRINT "♥";V=53248:SYS 49700
210 FOR N=0 TO 125:READ Q:POKE 49900+N,Q:NEXT
215 FOR I=2040 TO 2047:POKE I,13:NEXT
220 POKE V+23,255:POKE V+29,255
230 G=V:FOR I=0 TO 7:X(I)=G:G=G+1:Y(I)=G:G=G+1:NEXT
240 FOR I=0 TO 7:POKE X(I),120:POKE Y(I),120:NEXT
250 FOR I=0 TO 7:POKE V+39+I,1:NEXT
260 !VAL1,0:POKE V+21,255
270 !RUN 0,2,5::!RUN 1,1,8:!RUN 2,3,10::!RUN 3,4,0
280 !RUN 4,6,255::!RUN 5,7,35::!RUN 6,8,20::!RUN 7,9,4
290 FOR I=0 TO 15000:NEXT
300 FOR I=0 TO 7:!STOP I:NEXT
310 FOR I=0 TO 500:NEXT :POKE V+21,1:R=8
320 POKE X(0),200:POKE Y(0),200
325 !RUN 0,R,50
330 FOR I=0 TO 200:NEXT :!VAL1,1
340 FOR I=0 TO 200:NEXT :!VAL1,0
350 IF PEEK(Y(0))<50 OR PEEK(Y(0))>200 THEN R=10-R
360 GOTO 325
1000 DATA ,,,,,32,32,32,16,80,64,8,112,128,7,255,,1,252,,,136,,1,140
1010 DATA ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
2000 DATA ,,,,,,,32,,,80,,,112,,127,255,240,1,252,,,136,,1,140
2010 DATA ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
```

Bild 4. Dieses Programm dient als Beispiel für Sprites mit Bewegungseffekt

Zahl 5 kann eingegeben werden, hat aber keinen Einfluß. Bei der Geschwindigkeit G bedeutet 0 die schnellste und 255 die langsamste Bewegung. Die Sprites bewegen sich nun immer in der eingegebenen Geschwindigkeit. Sie können in Ihrem Basic-Programm berechnen oder abfragen so viel Sie wollen, dem Bewegungsdrang der Sprites tut dies keinen Abbruch. Verläßt ein Sprite den Bereich, in dem es sich bewegen kann (siehe Bild 1), so erscheint es wieder auf der gegenüberliegenden Seite. Wenn Sie dies nicht möchten, so müssen Sie durch Abfrage der entsprechenden Speicherstellen darauf reagieren. Anhalten können Sie ein Sprite mit dem Befehl
!STOP S.

Aufpassen muß man bei dem Befehl THEN. Zwischen THEN und dem neuen Befehl muß ein Doppelpunkt stehen, da es sonst zu einem SYNTAX-ERROR kommt.

Aufbau einer Spritebibliothek

Weiterhin haben Sie die Möglichkeit, ab der Speicherstelle 49900 bis zu 50 Sprites zu speichern und mit dem Befehl
!VAL B1,B2
in den Block zu schieben, auf dem Ihr Sprite zugreift. B1 kann dabei Werte von 0 bis 7 annehmen. Und zwar bedeutet:

0 = Block 11
ab Speicherstelle 704
1 = Block 13
ab Speicherstelle 832
2 = Block 14
ab Speicherstelle 896
3 = Block 15
ab Speicherstelle 960
4 = Block 32
ab Speicherstelle 2048
5 = Block 33
ab Speicherstelle 2112
6 = Block 34
ab Speicherstelle 2176
7 = Block 35
ab Speicherstelle 2240

Wenn Sie die Blöcke 32 bis 35 benützen, müssen Sie den Zeiger, der auf den Beginn des Basic-Programms zeigt, ändern. Mit POKE 2560,0 beginnt Ihr Basic-RAM-Bereich ab Speicherstelle 2561. Anschließend geben Sie noch NEW ein.

Bringen Sie Bewegung in die Sprites

B2 kann Werte von 0 bis 49 annehmen. Errechnen können Sie die Anfangsspeicherstelle für B2 mit 49900 + B2 * 63. Sie können sich so eine ganze Sprite-Bibliothek anlegen und bei Bedarf abrufen. In Bild 3 sehen Sie ein Programm, mit dem Sie die Sprites (die ab Speicherstelle 49900 stehen) zusammen mit dem Programm SPRITE-MOVE auf Kassette wegspeichern können. Denkbar wäre auch, daß man sich ähnlich aussehende Sprites speichert und diese abwechselnd einschaltet. Dadurch kann man den Eindruck einer sich bewegenden Figur erzeugen. (Siehe Beispielprogramm in Bild 4).

Alle Parameter für die neuen Befehle können Zahlen oder auch Variablen sein. Außerdem funktionieren die Befehle sowohl im Direktmodus als auch innerhalb eines Basicprogramms.

Sprites for ever

Das Programm steht von Speicherstelle 49229 bis 49798. Die Speicherstellen 49152 bis 49228 werden vom Programm als Speicher benutzt. Aktiviert werden die neuen Befehle mit SYS 49700. Im Basic-Lader geschieht dies in Zeile 172. Sollten Sie die RUN/STOP- und RESTORE-Taste betätigen, so müssen Sie das Programm neu aktivieren. Und nun viel Spaß mit den neuen, schnellen Sprites.

(Herbert Kunz)

Basic-Lader von »Sprite-Move« ►

```
100 REM  ******* SPRITE-MOVE *******
101 DATA 162,0,189,0,192,201,1,240,8
102 DATA 232,224,8,208,244,76,76,193
103 DATA 56,189,26,192,233,1,157,26,192
104 DATA 144,3,76,86,192,189,16,192,157
105 DATA 26,192,142,24,192,189,8,192
106 DATA 201,1,240,31,201,2,240,42,201
107 DATA 3,240,50,201,4,240,61,201,6
108 DATA 240,69,201,7,240,77,201,8,240
109 DATA 88,201,9,240,96,76,86,192,32
110 DATA 7,193,32,26,193,32,39,193,174
111 DATA 24,192,76,86,192,32,7,193,32
112 DATA 39,193,174,24,192,76,86,192
113 DATA 32,7,193,32,13,193,32,39,193
114 DATA 174,24,192,76,86,192,32,7,193
115 DATA 32,26,193,174,24,192,76,86,192
116 DATA 32,7,193,32,13,193,174,24,192
117 DATA 76,86,192,32,7,193,32,26,193
118 DATA 32,48,193,174,24,192,76,86,192
119 DATA 32,7,193,32,48,193,174,24,192
120 DATA 76,86,192,32,7,193,32,13,193
121 DATA 32,48,193,174,24,192,76,86,192
122 DATA 173,24,192,10,168,96,24,177
123 DATA 247,105,1,145,247,144,3,76,57
124 DATA 193,96,56,177,247,233,1,145
125 DATA 247,176,3,76,57,193,96,200,24
126 DATA 177,247,105,1,145,247,96,200
127 DATA 56,177,247,233,1,145,247,96
128 DATA 174,24,192,169,1,202,48,4,10
129 DATA 76,62,193,77,16,208,141,16,208
130 DATA 96,206,40,192,48,5,162,0,76
131 DATA 79,192,169,6,141,40,192,76,49
132 DATA 234,32,115,0,201,33,240,9,32
133 DATA 121,0,76,231,167,76,8,175,32
134 DATA 115,0,201,138,240,29,201,144
135 DATA 240,7,201,197,240,81,76,107
136 DATA 193,32,115,0,32,158,183,224
137 DATA 8,176,225,169,0,157,0,192,76
138 DATA 174,167,32,115,0,32,158,183
139 DATA 224,8,176,207,169,1,157,0,192
140 DATA 142,41,192,32,253,174,32,158
141 DATA 183,224,10,176,189,138,240,186
142 DATA 201,5,240,6,174,41,192,157,8
143 DATA 192,32,253,174,32,158,183,138
144 DATA 174,41,192,157,16,192,157,26
145 DATA 192,76,174,167,32,115,0,32,158
146 DATA 183,224,8,144,3,76,8,175,142
147 DATA 47,192,32,253,174,32,158,183
148 DATA 224,50,176,240,232,142,46,192
149 DATA 173,47,192,10,170,189,48,192
150 DATA 133,251,232,189,48,192,133,252
151 DATA 32,126,194,173,46,192,170,202
152 DATA 240,14,24,165,249,105,63,133
153 DATA 249,144,244,230,250,76,3,194
154 DATA 160,0,177,249,145,251,200,192
155 DATA 64,208,247,76,174,167,234,234
156 DATA 169,94,141,8,3,169,193,141,9
157 DATA 3,169,77,141,20,3,169,192,141
158 DATA 21,3,169,0,133,247,169,208,133
159 DATA 248,169,192,141,48,192,141,54
160 DATA 192,141,62,192,169,2,141,49
161 DATA 192,169,64,141,50,192,141,58
162 DATA 192,169,3,141,51,192,141,53
163 DATA 192,141,55,192,169,128,141,52
164 DATA 192,141,60,192,169,0,141,56
165 DATA 192,169,8,141,57,192,141,59
166 DATA 192,141,61,192,141,63,192,169
167 DATA 236,133,249,169,194,133,250
168 DATA 96
169 S=0:FOR I=49229 TO 49798:READ D
170 POKE I,D:S=S+D:NEXT
171 IF S<>69859 THEN PRINT "QFEHLER■":STOP
172 SYS 49700
```

BESTSELLER AUS BESTER HAND

Insgesamt über 200 000mal wurden die nachfolgenden Bücher in nur 12 Monaten verkauft.
Machen auch Sie mehr aus Ihrem COMMODORE mit diesen beliebten und bewährten Bestsellern aus bester Hand.

SIMON's BASIC ist ein Hit – wenn man es richtig nutzen kann. Deshalb gibt es jetzt zu dieser vielseitigen Befehlserweiterung unser umfangreiches Trainingsbuch, das Ihnen detailliert den Umgang mit den über 100 Befehlen des SIMON's BASIC erklärt. Ausführliche Darstellung aller Befehle (auch der, die nicht im Handbuch stehen!) Natürlich auch mit allen Macken und Hinweisen, wie man diese umgeht. Dazu zahlreiche Beispielprogramme und interessante Programmiertricks. Nach jedem Kapitel Testaufgaben zum optimalen Selbststudium. Dieses Buch sollte jeder SIMON's BASIC Anwender unbedingt haben! Das TRAININGSBUCH ZUM SIMON's BASIC, 1984, ca. 300 Seiten, DM 49,–.





Wer besser und leichter in BASIC programmieren möchte, der braucht dieses neue Buch. 64 FÜR PROFIS zeigt, wie man erfolgreich Anwendungsprobleme in BASIC löst und verrät Erfolgsgeheimnisse der Programmierprofis. Vom Programmentwurf über Menüsteuerung, Maskenaufbau, Parameterisierung, Datenzugriff und Druckausgabe bis hin zur Dokumentation wird anschaulich mit Beispielen dargelegt, wie gute BASIC-Programmierung vor sich geht. Fünf komplett beschriebene, lauffertige Anwendungsprogramme für den C-64 illustrieren den Inhalt der einzelnen Kapitel beispielhaft. Mit 64 FÜR PROFIS lernen Sie gute und erfolgreiche BASIC-Programmierung. 64 FÜR PROFIS, 1983, ca. 220 Seiten, DM 49,–.

Darauf haben Sie gewartet: Endlich ein Buch, das Ihnen ausführlich und verständlich die Arbeit mit der Floppy VC-1541 erklärt. DAS GROSSE FLOPPY BUCH ist für Anfänger, Fortgeschrittene und Profis gleichermaßen interessant. Sein Inhalt reicht von der Programmspeicherung bis zum DOS-Zugriff, von der sequentiellen Datenspeicherung bis zum Direktzugriff, von der technischen Beschreibung bis zum ausführlich dokumentierten DOS Listing, von den Systembefehlen bis zur detaillierten Beschreibung der Programme der Test/Demodiskette. Exakt beschriebene Beispiel- und Hilfsprogramme ergänzen dieses neue Superbuch. Mit dem GROSSEN FLOPPY-BUCH meistern Sie auch Ihre Floppy. DAS GROSSE FLOPPY BUCH, 1983, ca. 320 Seiten, DM 49,–.

Die überarbeitete und erweiterte 2. Auflage von 64 TIPS & TRICKS enthält eine umfangreiche Sammlung von POKE's und anderen nützlichen Routinen, Multitasking mit dem C-64, hochauflösende Graphik und Farbe für Fortgeschrittene, mehr über CP/M auf dem C-64, mehr über Anschluß- und Erweiterungsmöglichkeiten durch USER PORT und EXPANSION PORT, sowie zahlreiche ausführlich dokumentierte Programme von der SORT-Routine über zahlreiche BASIC-Erweiterungen bis hin zur 3D-Graphik (alle Maschinenprogramme jetzt mit BASIC-Ladeprogramm!). 64 TIPS UND TRICKS ist eine echte Fundgrube für jeden COMMODORE 64 Anwender. 64 TIPS & TRICKS, 2. Auflage 1983, ca. 290 Seiten, DM 49,–.

Jetzt in überarbeiteter und erweiterter 3. Auflage: 64 INTERN erklärt detailliert Architektur und technische Möglichkeiten des C-64, zerlegt mit einem ausführlich dokumentierten ROM-Listing Betriebssystem und BASIC-Interpreter, bringt mehr über Funktion und Programmierung des neuen Synthesizer Sound Chip und der hochauflösenden Graphik, zeigt die Unterschiede zwischen VC-20, C-64 und CBM 8000 und gibt Hinweise zur Umsetzung von Programmen. Zahlreiche lauffertige Beispielprogramme, Schaltbilder und als Clou: zwei ausführlich dokumentierte Original COMMODORE Schaltpläne zum Ausklappen. Dieses Buch sollte jeder 64-Anwender und Interessent haben. 64 INTERN, 3. Auflage 1983, ca. 320 Seiten, DM 69,–.





Die überarbeitete und erweiterte 2. Auflage von VC-20 INTERN beschäftigt sich detailliert mit Technik und Betriebssystem des VC-20 und enthält ein ausführlich dokumentiertes ROM-Listing, die Belegung der ZEROPAGE und anderer wichtiger Bereiche, übersichtliche Zusammenfassungen der Routinen des BASIC-Interpreters und des VC-20 Betriebssystems, eine Einführung in die Programmierung in Maschinensprache, eine detaillierte Beschreibung der Technik des VC-20 und als Clou drei Original COMMODORE Schaltpläne zum Ausklappen! Damit ist VC-20 INTERN für jeden interessant, der sich näher mit Technik und Maschinenprogrammierung des VC-20 auseinandersetzen möchte. VC-20 INTERN, 2. Auflage 1983, ca. 230 Seiten, DM 49,–.

Die überarbeitete und erweiterte 2. Auflage von VC-20 TIPS & TRICKS enthält eine detaillierte Beschreibung der Programmierung von Sound und Graphik des VC-20, mehr über Speicherbelegung, Speichererweiterung und die optimale Nutzung der einzelnen Speichermodule, BASIC-Erweiterungen zum Eintippen, umfangreiche Sammlung von Poke's und anderen nützlichen Routinen, zahlreiche interessante Beispiel- und Anwendungsprogramme, komplett dokumentiert und fertig zum Eintippen (z. B. Spiele, Funktionenplotter, Graphik Editor, Sound Editor) und vieles andere mehr. VC-20 TIPS & TRICKS ist eine echte Fundgrube für jeden VC-20 Anwender. VC-20 TIPS & TRICKS, 2. Auflage 1983, ca. 230 Seiten, DM 49,–.





Eine leicht verständliche Einführung in die Programmierung des COMMODORE 64 in Maschinensprache und Assembler für alle diejenigen, denen die Programmierung in BASIC nicht mehr ausreicht. Beispiele erläutern jeden neuen Befehl. Zur komfortablen Eingabe und zum Austesten Ihrer Maschinenprogramme enthält das Buch einen kompletten Assembler, einen Disassembler und einen Einzelschritt-Simulator, der besonders für den Anfänger sehr nützlich ist. Natürlich zugeschnitten auf Ihren Computer, den COMMODORE 64. DAS MASCHINENSPRACHEBUCH ZUM COMMODORE 64, 1984, ca. 200 Seiten, DM 39,–.

IHR GROSSER PARTNER FÜR KLEINE COMPUTER

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (02 11) 31 00 10 · im Hause AUTO BECKER

DATA BECKER BÜCHER und PROGRAMME erhalten Sie im Computer-Fachhandel, in den Computerabteilungen der Kauf- und Warenhäuser und im Buchhandel. Auslieferung für Österreich Fachbuchcenter ERB, Schweiz THALI AG und Benelux COMPUTERCOLLECTIEF.

BESTELL-COUPON H

Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1

Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten

☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

☐ DATA WELT 1/84 (DM 4,– in Briefmarken liegen bei)

Name und Adresse bitte deutlich schreiben

Invaders

Invaders ist eine vereinfachte, in Blockgrafik geschriebene Version des bekannten Apple-Invaders. Es benötigt eine Floppy-Disk und Simons Basic. Sie müssen die Erde gegen 66 Invader verteidigen. Werden Sie von einer Bombe getroffen oder erreicht ein Invader die Erde, so haben Sie Ihr Ziel, die Erde von den Eindringlingen zu befreien, nicht erreicht.

```
            INVADERS
           -========-

SIE SIND KOMMANDANT EINER BASIS
AUF DER ERDE. EINE FEINDLICHE KULTUR
HAT BESCHLOSSEN DIE RUECKSTAENDIGE
ERDE ZUM SCHUTZ DES WELTALL'S
ZU VERNICHTEN !

ALS VERTRETER DER IMPERIALISTISCHEN
ERDE VERTEIDIGEN SIE IHRE HEIMAT.

FUER JEDEN ABGESCHOSSENEN INVADER
ERHALTEN SIE PUNKTE .

     STEUERUNG:
  [1] BASIS LINKS
  [2] FEUER
  [3] BASIS RECHTS

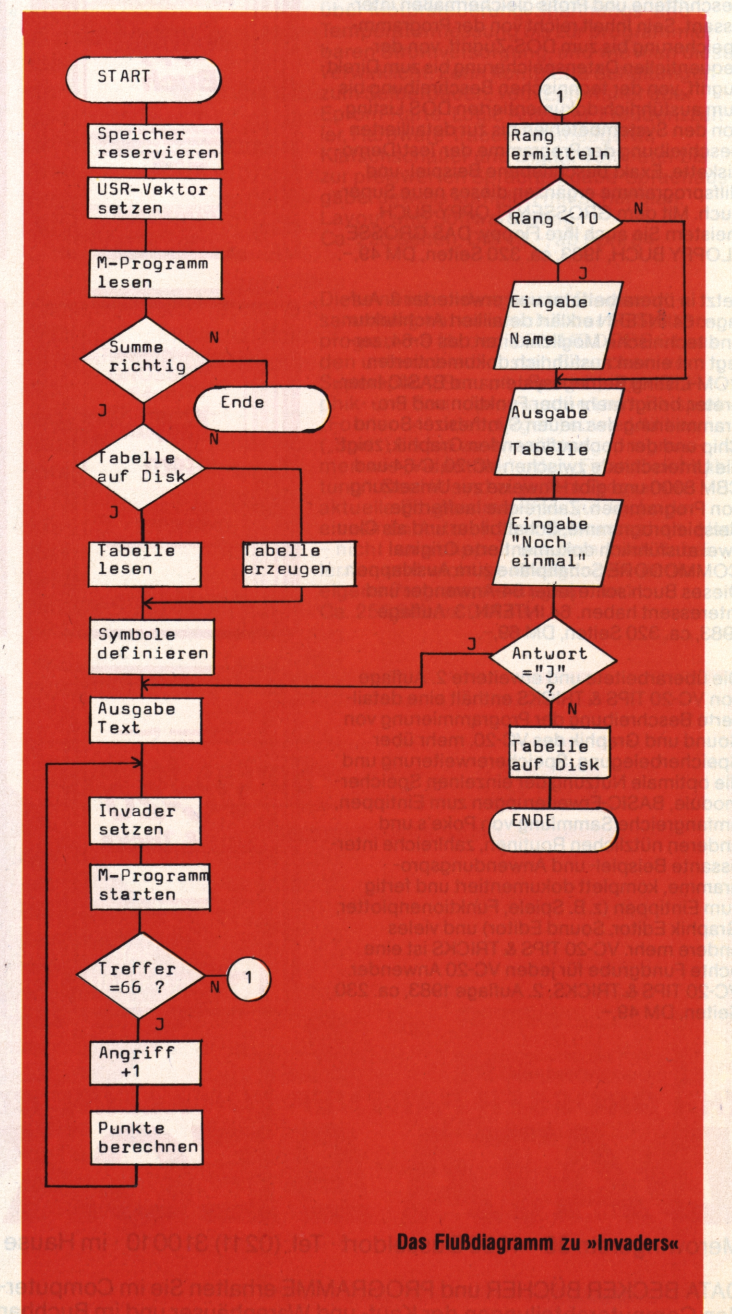
START?
```

Eine kurze Spielanleitung für »Invaders«

Block	Bedeutung
1200	Reserviert Speicherplatz und setzt USR-Vektor
1210-1250	M-Programm einlesen und Summe prüfen
1370-1420	Setze 66 Invader
1430	Setze Farben
1440	Aufruf des Maschinen-Programms
1500-1550	Punkte berechnen und neues Bild
1680-2580	Definiere Symbole
2680-2840	Titelbild drucken
2920-2940	Tabelle suchen
2950-2980	Tabelle lesen
2990-3020	Tabelle erzeugen
3040-3090	Rang ermitteln
3100-3130	Tabelle verschieben
3140-3160	Namen einfügen
3170-3220	Tabelle drucken
3230-3260	Abfrage »noch einmal?«
3270-3320	Tabelle auf Disk schreiben

Variable	Bedeutung
INDEX	Laufvariable allgemein
ZEILE	Laufvariable für Bildaufbau
SPALTE	Laufvariable für Bildaufbau
WERT	gelesener Datenwert
SUMME	Prüfsumme der Daten
ATTACK	Nummer des Angriffs
PUNKTE	Gesamtpunktzahl
TREFFER	Anzahl der getroffenen Invader
RANG	Tabellenposition
FEHLER	Fehlermeldung vom Disk
PUNKTE(I)	Punktetabelle
NAME§(I)	Namentabelle
A$	Abfrage allgemein

Die einzelnen Programmblöcke und Variablen mit ihrer Bedeutung



Das Flußdiagramm zu »Invaders«

So sieht »Invaders« auf dem Bildschirm aus. Die Farben kann man selbst bestimmen.

Da das Programm selbst gut dokumentiert und strukturiert ist, beschränke ich mich auf die Erklärung der Zeile 1430 (FCOL 0,0,40,25,1) und dem Aufruf TREFFER = USR (0) in Zeile 1440. Der FCOL-Befehl bewirkt, daß jedes Zeichen auf dem Bildschirm weiß erscheint. Wer das Spiel gern farbig haben möchte, muß daher nun an dieser Stelle Teilbereichen des Bildschirms mit FCOL eine andere Farbe zuweisen. Da das Programm voll verschiebbar ist, kann man sich dafür mit RENUMBER beliebig viel Platz verschaffen.

Grau raus — Farben rein

Der Aufruf TREFFER =USR(0) ruft das Maschinenprogramm ohne Zeile 3400 auf. Die Null hat dabei keine Bedeutung. Das Maschinenprogramm steuert die Basis, den Schuß, die Bomben und die 66 Invader. Wurde die Basis von einer Bombe getroffen oder hat ein Invader die Erde erreicht, wird der Variablen TREFFER die Anzahl der getroffenen Invader übergeben. Dann berechnet das Basic-Programm daraus die Punkte. Daher erhält man für jeden Invader die gleiche Punktzahl.

Zur Definition der Invader: Da das Programm in Blockgrafik geschrieben ist, werden alle verwendeten Symbole durch Umdefinieren von Zeichen erzeugt. Dabei ist eine Besonderheit zu beachten: Das Programm arbeitet mit drei verschiedenen Invader-Symbolen (Invader1 bis Invader3 in Zeile 1680 bis 2210). Jedes dieser Symbole ist in zwei Symbole A und B eingeteilt, zwischen denen das Maschinenprogramm bei jedem Schritt umschaltet.

(Manfred Friese)

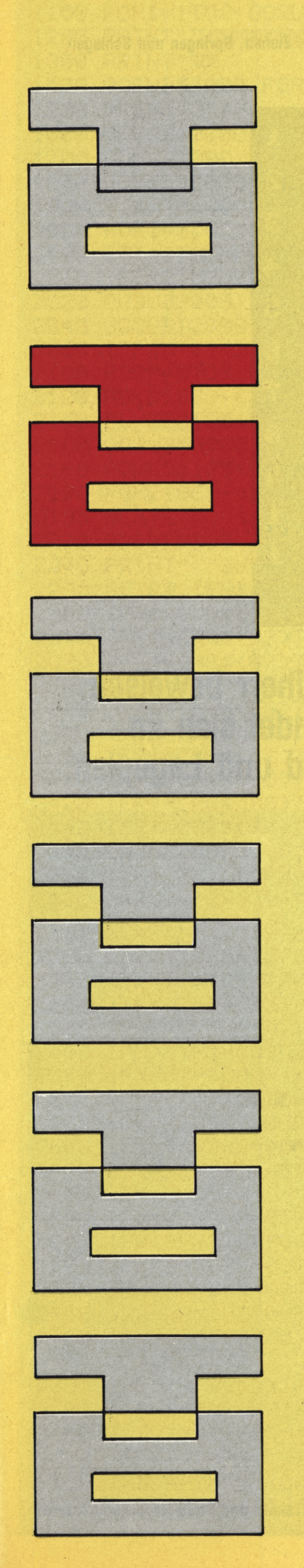
```
1000 REM*********************************
1010 REM***     I N V A D E R S       ***
1020 REM***                           ***
1030 REM***     FUER C 64 + 1541      ***
1040 REM***                           ***
1050 REM***    (C) M.FRIESE 1983      ***
1060 REM*********************************
1070 :
1080 :
1090 REM  "■" = CURSOR DOWN
1100 REM  "■" = CLEAR HOME
1110 REM  "■" = HOME
1120 REM  "■" = REVERS ON
1130 REM  "■" = REVERS OFF
1140 :
1150 :
1160 REM*********************************
1170 REM*** COMPUTER INITIALISIEREN ***
1180 REM*********************************
1190 :
1200 POKE56,124:CLR:POKE785,0:POKE786,124:POKE53281,0:PRINTCHR$(5)
1210 FOR  INDEX=0 TO 910
1220 :      READ WERT:POKE31744+INDEX,WERT
1230 :      SUMME=SUMME+WERT
1240 NEXT INDEX
1250 IF SUMME<>104201 THEN PRINT"DATEN FEHLER !":END
1260 EXEC READ TABLE
1270 EXEC DEF FIGURE
1280 :
1290 REM*********************************
1300 REM***       HAUPTPROGRAMM       ***
1310 REM*********************************
1320 :
1330 EXEC TEXT
1340 PROC NEU
1350 PUNKTE=0:ATTACK=1
1360 PROC START
1370 PRINT"■"
1380 FOR ZEILE=2 TO 12 STEP2
1390 :    FOR SPALTE=0 TO 20 STEP2
1400 :          FILL ZEILE+ATTACK-1,SPALTE,1,1,64+INT((ZEILE-1)/4)*2,1
1410 :    NEXT SPALTE
1420 NEXT ZEILE
1430 FCOL0,0,40,25,1
1440 TREFFER=USR(0):IF TREFFER=66 THEN CALL UEBERLEBT
1450 PUNKTE=PUNKTE+TREFFER*10↑ATTACK
1460 CALL TOT
1470 :
1480 :
1490 :
1500 PROC UEBERLEBT
1510 PRINT"■■■■■      ANGRIFF";ATTACK;"BEENDET■■■■"
1520 PUNKTE=PUNKTE+1234*10↑(ATTACK-1)
1530 PRINT"SIE HABEN";PUNKTE;"PUNKTE"
1540 PAUSE "■■■■WEITER MIT <RETURN> ",9999
1550 ATTACK=ATTACK+1:CALL START
1560 :
1570 :
1580 :
1590 PROC DEF FIGURE
1600 :
1610 REM*********************************
1620 REM*** DEFINIERT ALLE IM         ***
1630 REM*** PROGRAMM VERWENDETEN      ***
1640 REM*** SYMBOLE (BASIS,INVADER,   ***
1650 REM*** EXPLOSION,SCHUSS,BOMBE)   ***
1660 REM*********************************
1670 :
```

Listing des Simons Basic-Programms »Invaders« ►

Invaders

```
1680 MEM:DESIGN2,$E000+8*64:REM INVADER 1A
1690 @...BB...
1700 @..BBBB..
1710 @.BBBBBB.
1720 @BBBBBBBB
1730 @.BBBBBB.
1740 @..B..B..
1750 @.B....B.
1760 @B......B
1770 DESIGN2,$E000+8*65    :REM INVADER 1B
1780 @...BB...
1790 @..B..B..
1800 @.B....B.
1810 @B......B
1820 @.BBBBBB.
1830 @..B..B..
1840 @..B..B..
1850 @.B....B.
1860 DESIGN2,$E000+8*66    :REM INVADER 2A
1870 @........
1880 @BBBBBBBB
1890 @B.BBBB.B
1900 @BBBBBBBB
1910 @BB....BB
1920 @B.BBBB.B
1930 @BBBBBBBB
1940 @........
1950 DESIGN2,$E000+8*67    :REM INVADER 2B
1960 @........
1970 @BBBBBBBB
1980 @B.BBBB.B
1990 @BBB..BBB
2000 @BBBBBBBB
2010 @BB....BB
2020 @BBBBBBBB
2030 @........
2040 DESIGN2,$E000+8*68    :REM INVADER 3A
2050 @..B..B..
2060 @B..BB..B
2070 @B.BBBB.B
2080 @BBBBBBBB
2090 @..BBBB..
2100 @..BBBB..
2110 @.B....B.
2120 @B......B
2130 DESIGN2,$E000+8*69    :REM INVADER 3B
2140 @..B..B..
2150 @...BB...
2160 @..BBBB..
2170 @BBBBBBBB
2180 @B.BBBB.B
2190 @B.BBBB.B
2200 @..B..B..
2210 @.B....B.
2220 DESIGN2,$E000+8*70    :REM BASIS
2230 @........
2240 @........
2250 @........
2260 @...BB...
2270 @...BB...
2280 @.BBBBBB.
2290 @BBBBBBBB
2300 @BBBBBBBB
2310 DESIGN2,$E000+8*71    :REM BOMBE
2320 @B.B..B.B
2330 @.B.BB.B.
2340 @........
2350 @..B..B..
2360 @........
2370 @...BB...
2380 @..BBBB..
2390 @...BB...
2400 DESIGN2,$E000+8*72    :REM SCHUSS
2410 @...B....
2420 @...B....
2430 @..BBB...
2440 @..BBB...
2450 @..BBB...
2460 @..BBB...
2470 @..BBB...
2480 @..BBB...
2490 DESIGN2,$E000+8*73    :REM EXPLOSION
2500 @B..B...B
2510 @.B.B..B.
2520 @..B..B..
2530 @......BB
2540 @BB......
2550 @..B..B..
2560 @.B..B.B.
2570 @B...B..B
2580 END PROC
2590 :
2600 :
2610 :
2620 PROC TEXT
2630 :
2640 REM********************************
2650 REM* DRUCKT TITELBILD/ANLEITUNG   *
2660 REM********************************
2670 :
2680 PRINT"█           INVADERS"
2690 PRINT"           -=========-"
2700 PRINT"█SIE SIND KOMMANDANT EINER BASIS"
2710 PRINT"AUF DER ERDE. EINE FEINDLICHE KULTUR"
2720 PRINT"HAT BESCHLOSSEN DIE RUECKSTAENDIGE"
2730 PRINT"ERDE ZUM SCHUTZ DES WELTALL'S"
2740 PRINT"ZU VERNICHTEN !"
2750 PRINT"█ALS VERTRETER DER IMPERIALISTISCHEN"
2760 PRINT"ERDE VERTEIDIGEN SIE IHRE HEIMAT."
2770 PRINT"█FUER JEDEN ABGESCHOSSENEN INVADER"
2780 PRINT"ERHALTEN SIE PUNKTE ."
2790 PRINT"██   STEUERUNG:":PRINT" [1] BASIS LINKS"
2800 PRINT" [2] FEUER":PRINT" [3] BASIS RECHTS"
2810 PRINT"██START?"
2820 PROC WARTE
2830 GETA$:IFA$<>"J"THEN CALL WARTE
2840 END PROC
2850 :
2860 :
2870 :
2880 REM********************************
2890 REM***    TABELLENVERWALTUNG    ***
2900 REM********************************
2910 :
2920 PROC READ TABLE
2930 OPEN15,8,15:OPEN2,8,2,"TAB INVADERS,S,R"
2940 INPUT#15,FEHLER:IF FEHLER=62 THEN CALL NO TABLE
2950 FOR INDEX=0 TO 9
2960 :   INPUT#2,PUNKTE(INDEX)
2970 :   INPUT#2,NAME$(INDEX)
2980 NEXT:CLOSE2:CLOSE15:END PROC
2990 PROC NO TABLE
3000 FOR INDEX=0 TO 9
3010 :   PUNKTE(INDEX)=500:NAME$(INDEX)="***"
3020 NEXT:CLOSE2:CLOSE15:END PROC
3030 :
3040 PROC TOT
3050 PRINT"████*** SIE ERREICHTEN" PUNKTE "PUNKTE ***":PAUSE5
3060 RANG=10
3070 FOR INDEX=0 TO 9
3080 IF PUNKTE(INDEX)<PUNKTE AND RANG=10 THEN RANG=INDEX
3090 NEXT:IF RANG=10 THEN CALL PRINT TABLE
3100 FOR INDEX=9 TO RANG STEP-1
3110 :   NAME$(INDEX+1)=NAME$(INDEX)
3120 :   PUNKTE(INDEX+1)=PUNKTE(INDEX)
3130 NEXT
3140 PRINT"████NAME (MAX.19) ";
```

Listing des Simons Basic-Programms »Invaders« (Fortsetzung)

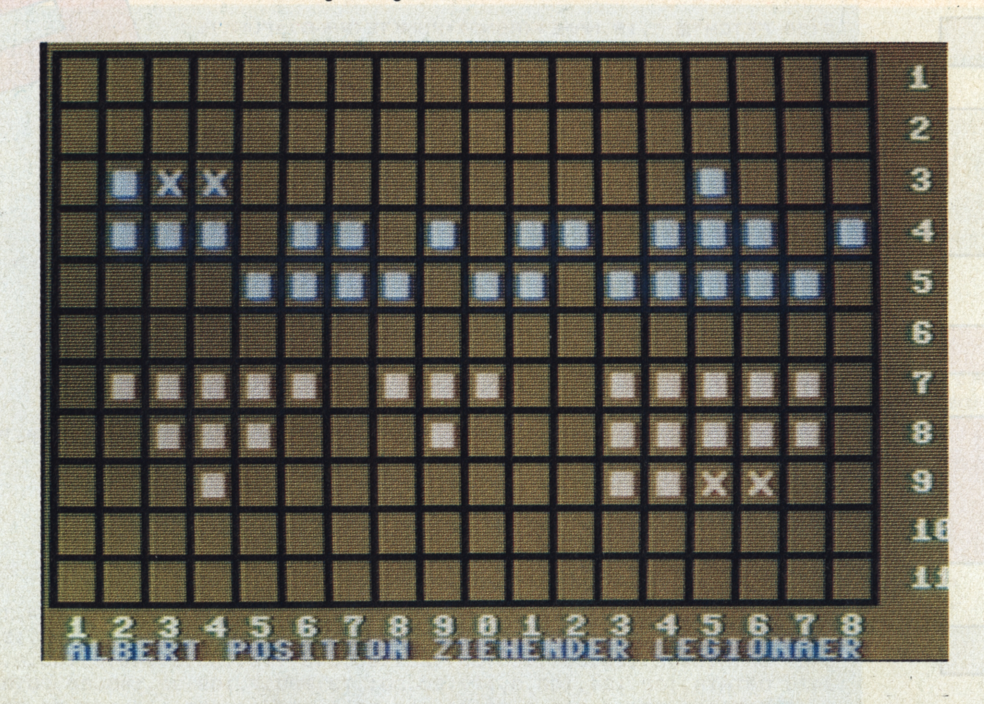


```
3150 FETCH"■ ",19,NAME$(RANG):PUNKTE(RANG)=PUNKTE
3160 IF NAME$(RANG)="" THEN NAME$(RANG)="***"
3170 PROC PRINT TABLE
3180 PRINT"■■*** REKORDE ***■■■"
3190 FOR INDEX=0 TO 9
3200 :    IF INDEX=RANG THENPRINT"■";
3210 :    PRINT INDEX+1,PUNKTE(INDEX),NAME$(INDEX);"■■"
3220 NEXT INDEX
3230 PRINT"■■NOCH EINMAL ?■ ";
3240 PROC WAIT
3250 GETA$:IF A$<>"J" AND A$<>"N" THEN CALL WAIT
3260 PRINTA$;:IF A$="J" THEN CALL NEU
3270 OPEN2,8,2,"@0:TAB INVADERS,S,W"
3280 FOR INDEX=0 TO 9
3290 :    PRINT#2,PUNKTE(INDEX)
3300 :    PRINT#2,NAME$(INDEX)
3310 NEXT
3320 CLOSE2
3330 :
3340 :
3350 :
3360 REM********************************
3370 REM***    MASCHINENPROGRAMM      ***
3380 REM********************************
3390 :
3400 DATA32,164,124,169,207,133,101,169,210,133,100,32,182,124,169,70
3410 DATA160,0,145,20,32,173,124,169,12,133,252,169,0,133,102,133,110,133,108
3420 DATA133,2,133,251,165,251,201,66,240,110,165,2,240,8,198,2,32,246,126
3430 DATA76,122,124,160,0,177,20,201,73,208,7,169,32,145,20,76,82,124,32,4
3440 DATA125,224,0,240,3,32,26,125,32,5,126,32,225,124,165,20,201,0,208,219
3450 DATA165,21,201,204,208,213,56,169,66,229,251,74,74,74,133,2,32,102,127
3460 DATA32,57,125,32,207,125,32,43,126,32,85,126,32,190,126,32,173,124,165
3470 DATA97,240,5,56,233,3,133,97,24,105,128,141,24,212,165,108,201,1,208,143
3480 DATA32,34,127,32,17,125,164,251,32,162,179,96,165,20,133,98,165,21,133
3490 DATA99,96,169,255,133,20,169,207,133,21,96,165,100,133,20,165,101,133
3500 DATA21,96,165,105,133,20,165,106,133,21,96,165,20,133,100,165,21,133,101
3510 DATA96,165,20,133,105,165,21,133,106,96,230,20,208,2,230,21,96,198,20
3520 DATA166,20,224,255,208,2,198,21,96,165,20,24,105,40,144,2,230,21,133,20
3530 DATA96,165,20,56,233,40,176,2,198,21,133,20,96,162,0,201,64,48,6,201,70
3540 DATA16,2,162,1,96,165,98,133,20,165,99,133,21,96,73,1,166,102,208,13,32
3550 DATA218,124,145,20,169,32,32,225,124,145,20,96,72,169,32,145,20,32,225
3560 DATA124,104,145,20,96,32,179,125,165,102,201,1,240,4,169,39,133,20,169
3570 DATA0,133,107,177,20,32,4,125,224,1,240,13,32,236,124,166,107,232,134
3580 DATA107,224,23,208,235,96,165,102,73,1,133,102,32,173,124,177,20,133,109
3590 DATA32,4,125,224,1,208,14,32,236,124,165,109,145,20,32,248,124,169,32
3600 DATA145,20,32,225,124,165,20,201,0,208,222,165,21,201,204,208,216,32,179
3610 DATA125,162,23,32,236,124,202,208,250,162,40,134,107,32,218,124,177,20
3620 DATA32,4,125,224,1,208,2,134,108,198,107,208,238,96,169,0,133,20,169,204
3630 DATA133,21,96,169,128,141,18,212,169,143,141,24,212,169,1,141,15,212,173
3640 DATA27,212,96,32,179,125,162,4,32,188,125,24,101,20,133,20,144,2,230,21
3650 DATA202,208,241,177,20,32,4,125,224,1,208,24,32,236,124,177,20,201,32
3660 DATA208,4,169,71,145,20,201,70,208,7,169,1,133,108,76,245,125,96,177,20
3670 DATA201,71,208,24,169,32,145,20,32,236,124,177,20,201,32,208,4,169,71
3680 DATA145,20,201,70,240,4,32,248,124,96,169,1,133,108,76,24,126,32,182,124
3690 DATA32,188,125,41,15,240,246,133,107,32,248,124,198,107,208,249,177,20
3700 DATA32,4,125,224,1,208,13,32,236,124,177,20,201,32,208,4,169,71,145,20
3710 DATA96,32,182,124,32,159,255,160,0,165,203,201,59,208,17,165,110,201,1
3720 DATA240,10,32,209,124,32,6,127,169,1,133,110,96,201,56,208,30,169,32,145
3730 DATA20,32,225,124,165,20,201,191,208,3,32,218,124,177,20,201,71,240,43
3740 DATA169,70,145,20,32,200,124,96,201,8,240,1,96,169,32,145,20,32,218,124
3750 DATA165,20,201,232,208,3,32,225,124,177,20,201,71,240,8,169,70,145,20
3760 DATA32,200,124,96,169,1,133,108,96,165,110,208,1,96,32,191,124,177,20
3770 DATA201,72,208,4,169,32,145,20,32,248,124,177,20,201,32,208,8,169,72,145
3780 DATA20,32,209,124,96,32,4,125,224,0,240,5,230,251,32,77,127,169,73,145
3790 DATA20,169,0,133,110,96,169,8,133,21,32,225,124,165,20,208,249,165,21
3800 DATA208,245,96,169,0,141,11,212,141,0,212,169,18,133,97,169,100,141,1
3810 DATA212,169,129,141,4,212,169,240,141,6,212,96,169,7,141,1,212,169,15
3820 DATA141,24,212,169,0,141,11,212,141,18,212,141,4,212,141,5,212,169,252
3830 DATA141,6,212,169,129,141,4,212,32,246,126,169,128,141,4,212,96,169,7
3840 DATA141,1,212,169,18,133,97,169,129,141,4,212,169,240,141,6,212,169,0
3850 DATA141,11,212,96,165,252,56,233,2,201,2,208,2,169,10,141,8,212,133,252
3860 DATA169,15,133,97,169,240,141,13,212,169,143,141,24,212,169,0,141,4,212
3870 DATA169,33,141,11,212,96
READY.
```

Listing des Simons Basic-Programms »Invaders« (Schluß)

Kämpfe wie im alten ROM

Dieses Bild gibt einen Ausschnitt aus dem Spielverlauf wieder. Ziel ist es, das Feldlager des Kontrahenten zu erreichen. Ziehen, Springen und Schlagen der Legionäre geschieht in ähnlicher Weise wie bei Dame.



Mit »Caesar« können Sie Ihr taktisches Geschick als Feldherr beweisen. Es ist ein Spiel für zwei Personen. Die Anleitung befindet sich im Programm. Der Commodore 64 dient Ihnen als Schlachtfeld und Lageplan.

```
100 DIMF(19,12):DIMD(2):T1=1229
120 PRINT"♥"
140 SD=1
160 SI=54272:FL=SI:FH=SI+1:TL=SI+2:TH=SI+3
180 W9=SI+4:A=SI+5:H=SI+6:L=SI+24
200 F(3,3)=5:F(4,3)=5:T2=1231
220 F(15,9)=6:F(16,9)=6:T3=56005
240 D(1)=25:D(2)=25:T4=56007
420 POKE53280,0:POKE53281,0
440 PRINT"▒▒▒▒▒▒▒▒▒■"
460 PRINT"  ◤   ▀  ◤   ▀  ◤   ▀  ◤   ▀  ◤   ▀"
480 PRINT"  ▌▀     ▌▀   ▌▀ ▌▀     ▌▀     ▌▀  ▌▀ ▌▀   ▌▀"
500 PRINT"  ▌▀     ▌      ▀ ▌     ▀ ▌     ▀ ▌     ▀ ▌     ▀"
520 PRINT"  ▌▀     ▌▀   ▌▀ ▌▀       ▌▀ ▌▀   ▌▀ ▌▀ ◥"
540 PRINT"  ◥▌    ▄ ▌▀   ▌▀◥▌    ▄◥▌    ▄ ▌▀   ▌▀◥   ◥ "
580 POKE53281,1:FORT=1TO1500:NEXTT
600 DATA10,89,200,16,109,200,27,160,200
620 DATA34,207,200,-1,-1,0
640 DATA15,129,70,15,129,70,13,10,70
660 DATA29,69,100,-1,-1,0
680 DATA5,123,200,17,103,200,14,162,200
700 DATA87,182,270,-1,-1,0
720 DATA16,109,130,16,109,130
740 DATA16,109,150,3,116,270,-1,-1,0
760 DATA17,103,200,21,237,200,26,20
780 DATA330,21,237,80,26,20,200,29
800 DATA69,200,26,20,200,0,0,200,21
820 DATA237,200,26,20,200,29,69,600
840 DATA26,20,200,0,0,200
```

Listing des Taktik- und Strategiespiels »Caesar«

```
860 DATA-1,-1,0
880 GOSUB51000
900 PRINT"■"
1000 GOSUB40000
1100 FORI=1TO2:GOSUB10000:GOSUB15000:NEXTI:POKE53280,14:POKE53281,8
1200 PRINT"■■■■■■■■■■      OK. DIE LEGIONAERE SIND POSTIERT."
1300 PRINT"■■           ICH WERDE MAL AUFDECKEN."
1400 GOSUB54000:FORT=1TO1000:NEXTT
1500 PRINT"■";
1600 GOSUB[illegible]0000
1700 GOSUB7000
1800 GOSUB50000
1900 TI$="000000"
2000 FORI=1TO2:RR=0
2005 IFI=1THENPRINT"■";
2010 IFI=2THENPRINT"■";
2020 GOSUB9000:PRINT" "SN$(I)" POSITION ZIEHENDER LEGIONAER■":GOSUB50000
2040 GOSUB12000:GOSUB9000
2060 IFF(X,Y)<>ITHENGOSUB9000:PRINT"FALSCHE EINGABE";:GOSUB9700:GOTO2040
2100 FORP=0TO9
2120 POKE1863+(X*2)-((11-Y)*80),224
2140 POKE56135+(X*2)-((11-Y)*80),9
2160 GOSUB56000
2180 POKE56135+(X*2)-((11-Y)*80),1
2190 POKE1863+(X*2)-((11-Y)*80),86
2195 NEXTP
2200 GOSUB9000
2240 PRINT" "SN$(I)" WELCHE RICHTUNG?■"
2280 GETR$:IFR$=""THEN2280
2300 IFR$=CHR$(81)THENZX=-1:ZY=-1:GOTO2450
2310 IFR$=CHR$(83)THEN25000
2320 IFR$=CHR$(87)THENZX=0:ZY=-1:GOTO2450
2330 IFR$=CHR$(77)THEN34000
2340 IFR$=CHR$(69)THENZX=1:ZY=-1:GOTO2450
2360 IFR$=CHR$(65)THENZX=-1:ZY=0:GOTO2450
2380 IFR$=CHR$(68)THENZX=1:ZY=0:GOTO2450
2400 IFR$=CHR$(90)THENZX=-1:ZY=1:GOTO2450
2420 IFR$=CHR$(88)THENZX=0:ZY=1:GOTO2450
2440 IFR$=CHR$(67)THENZX=1:ZY=1:GOTO2450
2445 GOTO2280
2450 IFX+(ZX)<1ORX+(ZX)>18ORY+(ZY)<1ORY+(ZY)>11THEN22000
2500 W=F(X+(ZX),Y+(ZY)):GOSUB55000
2520 W2=F(X+(ZX*2),Y+(ZY*2))
2530 IFRR=1THEN2570
2540 IFW=I+4THEN22000
2560 IFW=0THENF(X,Y)=0:GOSUB20000:F(X+(ZX),Y+(ZY))=I:BX=X+ZX:BY=Y+ZY
2565 IFW=0THENF(X,Y)=0:GOSUB21000:GOTO6800
2566 IFW>4THENGOSUB20000:BX=X+ZX:BY=Y+ZY:GOTO6800
2570 IFW2=I+4ORW=I+4THEN22000
2575 IFX+(ZX*2)<1ORX+(ZX*2)>18ORY+(ZY*2)<1ORY+(ZY*2)>11THEN22000
2580 IFW=IANDW2=0ORW=IANDW2>3THENF(X,Y)=0:GOSUB20000
2600 IFW=IANDW2=0ORW=IANDW2>3THENF(X+(ZX*2),Y+(ZY*2))=I
2620 IFW=IANDW2=0ORW=IANDW2>3THENBX=X+(ZX*2):BY=Y+(ZY*2):GOSUB21000:GOTO6000
2640 REM
2660 IFW=0ANDW2=0THEN22000
2680 IFW<3ANDW2=0ORW<3ANDW2>3THENF(X,Y)=0:GOSUB20000:F(X+(ZX),Y+(ZY))=0
2700 IFW<3ANDW2=0ORW<3ANDW2>3THENBX=X+ZX:BY=Y+ZY:GOSUB23000
2720 IFW<3ANDW2=0ORW<3ANDW2>3THENF(X+(ZX*2),Y+(ZY*2))=I:BX=X+(ZX*2):BY=Y+(ZY*2)
2740 IFW<3ANDW2=0ORW<3ANDW2>3THENGOSUB21000:GOSUB53000:GOTO6000
2760 IFW>2GOTO6800
2800 GOTO24000
6000 IFF(BX,BY)>2THEN6800
6200 GOSUB9000:PRINT"  WEITERSPRINGEN? JA=F1 NEIN=F7";
6500 GETSS$:IFSS$=""THEN6500
6520 IFSS$=CHR$(133)THENX=X+(ZX*2):Y=Y+(ZY*2):ZX=0:ZY=0:RR=1:GOTO2100
6540 IFSS$=CHR$(134)THEN6800
6560 IFSS$=CHR$(135)THEN6800
```

Listing des Taktik- und Strategiespiels »Caesar« (Fortsetzung)

```
6580 IFSS$=CHR$(136)THEN6800
6600 GOTO6500
6800 IFBX=3ANDBY=3ANDI=2THENPOKE55501,14:POKET1,88:F(3,3)=3:L2=L2+1:D(2)=D(2)-1
6820 IFBX=4ANDBY=3ANDI=2THENPOKE55503,14:POKET2,88:F(4,3)=3:L2=L2+1:D(2)=D(2)-1
6840 IFBX=15ANDBY=9ANDI=1THENPOKET3,10:POKE1733,88:F(15,9)=3:L1=L1+1:D(1)=D(1)-1
6860 IFBX=16ANDBY=9ANDI=1THENPOKET4,10:POKE1735,88:F(16,9)=3:L1=L1+1:D(1)=D(1)-1
6880 IFL1=2ORL2=2THEN35000
6900 IFD(1)=0ORD(2)=0THEN38000
6920 GOSUB9000:NEXTI     I gelöscht
6940 RZ=RZ+1:I=0:GOTO2000
7000 FORXX=1TO18
7200 FORYY=1TO11
7300 IFF(XX,YY)=1THENGOSUB7600
7400 IFF(XX,YY)=2THENGOSUB7700
7450 REM    IFF(XX,YY)=3THENGOSUB7800
7500 NEXTYY,XX:XX=0:YY=0:RETURN
7600 POKE1863+(XX*2)-((11-YY)*80),224:POKE56135+(XX*2)-((11-YY)*80),14:RETURN
7700 POKE1863+(XX*2)-((11-YY)*80),224:POKE56135+(XX*2)-((11-YY)*80),10:RETURN
7800 REMPOKE1863+(XX*2)-((11-YY)*80),224:POKE56135+(XX*2)-((11-YY)*80),9:RETURN
8000 IFI=2THEN8200
8100 POKE1863+(X*2)-((11-Y)*80),224:POKE56135+(X*2)-((11-Y)*80),14:RETURN
8200 POKE1863+(X*2)-((11-Y)*80),224:POKE56135+(X*2)-((11-Y)*80),10:RETURN
9000 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"gemacht                                    ";
9100 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";
9200 RETURN
9500 GOSUB9000
9520 PRINT"█          NICHT MOEGLICHER ZUG█";
9540 GOSUB9700:GOSUB9000:GOTO2120
9600 GOSUB9000
9620 PRINT"█          U N M O E G L I C H █";
9640 GOSUB9700:GOSUB9000:GOTO2120
9700 FORHR=1TO1500:NEXTHR
9800 RETURN
10000 POKE53281,9:POKE53280,0
10100 PRINT"██ ┌┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬─┬┐"
10200 B1$="| | | | | | | | | | | | | | | | | | |"
10300 B2$="├─┼─┼─┼─┼─┼─┼─┼─┼─┼─┼─┼─┼─┼─┼─┼─┼─┼─┤"
10400 FORB=1TO10:PRINTB1$:PRINTB2$:NEXTB
10500 PRINTB1$:PRINT"└─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┴─┘"
10600 PRINT"█":FORB=1TO11:PRINTTAB(37)B;:PRINT:NEXTB
10700 POKE55501,14:POKE55503,14:POKE1229,86:POKE1231,86
10800 POKE56005,10:POKE56007,10:POKE1733,86:POKE1735,86
10900 PRINT" 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8":GOSUB11000
11000 FORB=56216TO56255:POKEB,13:NEXTB
11100 FORB=55334TO56294STEP40:POKEB,13:POKEB+1,13:NEXTB
11200 RETURN
12000 GETX$:IFX$>CHR$(47)ANDX$<CHR$(58)THEN12200
12100 GOTO12000
12200 X=VAL(X$):POKE56215+(X*2),0
12300 GOSUB9000:IFX>1THEN13000
12400 GETX$:IFX$>CHR$(47)ANDX$<CHR$(57)THEN12600
12450 IFX$=CHR$(13)THEN13000
12500 GOTO12400
12600 X=VAL(X$):X=X+10:POKE56215+(X*2),0
12700 REM
13000 GETY$:IFY$>CHR$(47)ANDY$<CHR$(58)THEN13200
13100 GOTO13000
13200 Y=VAL(Y$):POKE55294+(Y*80),0:POKE55295+(Y*80),0
13300 IFY>1THEN13900
13400 GETY$:IFY$>CHR$(47)ANDY$<CHR$(50)THEN13600
13450 IFY$=CHR$(13)THEN13900
13500 GOTO13400
13600 POKE55375,0:Y=VAL(Y$):Y=Y+10:POKE55294+(Y*80),0:POKE55295+(Y*80),0
13900 FORT=1TO500:NEXTT:GOSUB11000
14000 RETURN
15000 GOSUB9000
15100 PRINT" "SN$(I)" GEBE DEINE FIGUREN JETZT EIN";:GOSUB9700
```

Listing des Taktik- und Strategiespiels »Caesar« (Fortsetzung)

```
15300 FORS=1TO25:GOSUB9000:PRINT"■ "S". FIGUR";
15400 GOSUB12000:GOSUB9000
15600 IFF(X,Y)>0THENGOSUB9000:PRINT"■ UNMOEGLICH";:GOTO15400
15700 IFI=1ANDY>5THENGOSUB9000:PRINT"■ UNMOEGLICH";:GOTO15400
15800 IFI=2ANDY<7THENGOSUB9000:PRINT"■ UNMOEGLICH";:GOTO15400
15900 IFX=15ANDY=9THENGOSUB9000:PRINT"■ DAS IST DAS ZIELFELD";:GOTO15400
16000 IFX=3ANDY=3THENGOSUB9000:PRINT"■ DAS IST DAS ZIELFELD";:GOTO15400
16100 IFX=16ANDY=9THENGOSUB9000:PRINT"■■DAS IST DAS ZIELFELD";:GOTO15400
16200 IFX=4ANDY=3THENGOSUB9000:PRINT"■ DAS IST DAS ZIELFELD";:GOTO15400
16300 GOSUB9000
16400 F(X,Y)=I
16500 GOSUB8000
16600 NEXTS:IFI=2THEN17300
16700 FORT=1TO800:NEXTT
16800 GOSUB50000
16900 PRINT"■■■■■■■■■■CAESAR WUENSCHT JETZT DIE HEERESSTELLUNG■"
17000 PRINTTAB(11)"VON HEERESFUEHRER■■"
17050 PRINTTAB(14)SN$(2)
17100 FORT=1TO3500:NEXTT
17300 FORT=1TO750:NEXTT
17400 PRINT"■";:RETURN
20000 POKE1863+(X*2)-((11-Y)*80),224:POKE56135+(X*2)-((11-Y)*80),9:RETURN
21000 IFI=1THEN21300
21100 IFI=2THEN21500
21300 POKE1863+(BX*2)-((11-BY)*80),224:POKE56135+(BX*2)-((11-BY)*80),14:RETURN
21400 RETURN
21500 POKE1863+(BX*2)-((11-BY)*80),224:POKE56135+(BX*2)-((11-BY)*80),10:RETURN
21600 RETURN
22000 GOSUB9000
22100 PRINT"       CAESAR ERLAUBT DIES NICHT.";
22200 GOSUB9700
22300 GOSUB9000
22320 IFI=1THEN22400
22340 IFI=2THEN22600
22400 POKE1863+(X*2)-((11-Y)*80),224:POKE56135+(X*2)-((11-Y)*80),14:GOTO6800
22600 POKE1863+(X*2)-((11-Y)*80),224:POKE56135+(X*2)-((11-Y)*80),10:GOTO6800
23000 POKE1863+(BX*2)-((11-BY)*80),224:POKE56135+(BX*2)-((11-BY)*80),9:RETURN
24000 GOSUB9000
24100 PRINT" CAESAR DULDET KEINE MEUCHELMOERDER.";:GOTO22200
25000 GOSUB9700
25100 GOSUB9000
25200 IFI=1THEN25400
25300 IFI=2THEN25500
25400 POKE1863+(X*2)-((11-Y)*80),224:POKE56135+(X*2)-((11-Y)*80),14:GOTO6800
25500 POKE1863+(X*2)-((11-Y)*80),224:POKE56135+(X*2)-((11-Y)*80),10:GOTO6800
30000 PRINT"■■■■■■■"SPC(8)"SOLL ICH DIE STEUERUNGS-"
30100 PRINTTAB(10)"FUNKTIONEN NAEHER ER-"
30200 PRINTTAB(15)"LAEUTERN?"
30400 GETY3$:IFY3$=""THEN30400
30500 IFY3$="J"THEN30700
30600 RETURN
30700 PRINT"■■■":POKE53281,2:POKE53280,0
30800 PRINT"DIE ANSTEUERUNG JEDES EINZELNEN SOLDATEN";
30900 PRINT"  ERFOLGT UEBER EINGABE DER EINZELNEN"
31000 PRINT"KOORDINATEN DESSELBEN. WENN SIE ZB. DEN"
31100 PRINT" SOLDATEN MIT DER KOORDINATE 4 / 6 AN-"
31200 PRINT" STEUERN WOLLEN DRUECKEN SIE 4; DANN 6."
31300 PRINT"BEI EINEM SOLDATEN MIT DEN KOORDINATEN"
31400 PRINT"1 / 11 SIEHT DIE SACHE ETWAS ANDERS AUS;";
31500 PRINT"HIER GEBEN SIE EIN: 1 - RETURN - 1 - 1."
31600 PRINT"  NACHDEM SIE DIE POSITION EINGEGEBEN"
31700 PRINT" HABEN MUESSEN SIE NOCH DIE ZUGRICHTUNG"
31800 PRINT"  EINGEBEN.  DAS GESCHIEHT WIE FOLGT:■■"
31900 PRINT"          LINKS  OBEN   RECHTS"
32000 PRINT"              \  !  /       "
32100 PRINT"             ■Q■ ■W■ ■E■        "
32200 PRINT"              \!/       "
```

Listing des Taktik- und Strategiespiels »Caesar« (Fortsetzung)

```
32300 PRINT"            LINKS -▌A▐-●-▌D▐- RECHTS"
32400 PRINT"                         /|\         "
32500 PRINT"                        ▌Z▐ ▌X▐ ▌C▐       "
32600 PRINT"                       /  |  \       "
32700 PRINT"            LINKS  UNTEN  RECHTS"
32750 GETPQ$:IFPQ$=""THEN32750
32800 PRINT"█████WENN SIE ZWISCHENZEITIC AUFHOEREN WOLLEN";
32900 PRINT"DRUECKEN SIE ´M´ WENN SIE EINEN ZUG, DER";
33000 PRINT"BEREITS MIT KOORDINATEN EINGEGEBEN WURDE";
33100 PRINT"NICHT MACHEN WOLLEN, DRUECKEN SIE ´S´."
33200 PRINT"BEACHTEN SIE: SIE DUERFEN NICHT MIT EIG-";
33300 PRINT"ENEN LEGIONAEREN INS EIGENE LAGER. SIE"
33400 PRINT"SOLLEN NICHT VERSUCHEN AUS DEM SCHLACHT-";
33500 PRINT"FELD HERAUSZUZIEHEN UND WEITERHIN NICHT"
33600 PRINT"AUF SOLDATEN SPRINGEN - ALL DIES BE-"
33700 PRINT"STRAFT CAESER DADURCH INDEM DER NAECHSTE";
33800 PRINT"HEERFUEHRER AN DIE REIHE KOMMT."
33900 PRINT"██████████VMI█EML █S MP █A MS █S █████":GOTO30400
34000 GOSUB52000:GOSUB52000
34050 PRINT"████████":POKE53281,0:POKE53280,2
34100 PRINT"  CAESAR BEWUNDERT DIESE STRATEGISCHE"
34150 PRINT"  LEISTUNG NICHT BESONDERS. SCHON NACH"
34200 PRINT"  "RZ" ZUEGEN AUFZUHOEREN IST NICHT"
34300 PRINT"  EINES CAESARS HEERFUEHRER ANGEMESSEN."
34350 PRINT"  VIELLEICHT SOLLTET IHR ES EIN PAAR"
34400 PRINT"  JAHRHUNDERTE SPAETER BEI DEM EHRVOL-"
34450 PRINT"  LEN KAISER WILHELM NOCH EINMAL VER-"
34500 PRINT"  SUCHEN; ODER IHR KOENNTET ES BEI MIR"
34550 PRINT"  IN EINER MEINER ARENEN PROBIEREN."
34600 FORT=1TO4000:NEXTT:GOTO39000
35000 FORT=1TO900:NEXTT
35100 PRINT"█████████████████ ╭──────────────────────╮"
35150 PRINT"█████████│DER SIEGER STEHT FEST!│"
35200 PRINT"█████████╰──────────────────────╯"
35250 GOSUB54000
35300 PRINT"█████":POKE53280,0:POKE53281,2
35350 IFL1=2THENG$=SN$(1)
35370 IFL2=2THENG$=SN$(2)
35400 IFRZ<27THEN38000
35500 IFRZ<65THEN36300
35550 GOSUB52000:GOSUB52000
35600 PRINT" MEINEN GLUECKWUNSCH "G$";"
35650 PRINT" DEINE LEISTUNG HAETTE ZWAR WESENTLICH"
35700 PRINT" BESSER SEIN KOENNEN, DA DU "RZ/2
35750 PRINT" VERSCHIEBUNGEN DEINER HEERESFORMATION"
35800 PRINT" GEBRAUCHT HAST UM ZU GEWINNEN."
35850 PRINT" MIT EIN BISSCHEN UEBUNG SOLLTST DU "
35900 PRINT" DEINE LEISTUNG ABER NOCH STEIGERN"
35950 PRINT" KOENNEN. ZUR ZEIT KANN ICH DICH ABER"
36000 PRINT" NUR ALS VORKOSTER IN MEINEM STAB GE-"
36050 PRINT" BRAUCHEN; DA DER LETZTE, MERKWUERDIGER-";
36100 PRINT" WEISE,NACHDEM BRUTUS SEIN GASTGESCHENK;";
36150 PRINT" EINE SACHERTORTE; VORBEIBRACHTE; AN"
36200 PRINT" MAGENKRAEMPFEN VON UNS GEGANGEN IST."
36250 FORT=1TO4000:NEXTT:GOTO39000
36300 GOSUB54000:PRINT"   MEINEN GLUECKWUNSCH, "G$:GOSUB52000
36350 PRINT"█   DEINE ART DER HEERESFUEHRUNG"
36400 PRINT"   ERINNERT AN DIE VON NERO DEN"
36450 PRINT"   ICH ZWAR NICHT SO MAG, ABER ICH"
36500 PRINT"   GRATULIERE DIR TROTZDEM. DEINE"
36550 PRINT"   LEGIONAERE FINDEN, DAS DU"
36600 PRINT"   EIN BARSCHER, UNLIEBSAMER"
36650 PRINT"   FUEHRER BIST, UND, OFFEN"
36700 PRINT"   GESAGT, SIE HASSEN DICH."
36750 PRINT"   ZWAR KANN ICH DIR ZUR ZEIT"
36800 PRINT"   KEINEN POSTEN ALS HEERES-"
36850 PRINT"   FUEHRER IN MEINER LEGION"
```

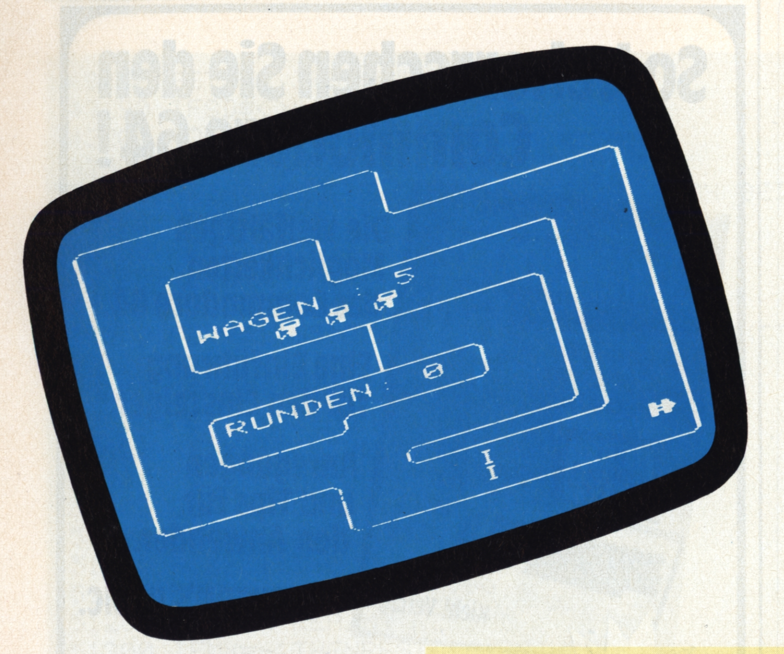
Listing des Taktik- und Strategiespiels »Caesar« (Fortsetzung)

```
36900 PRINT"   ANBIETEN, ABER EIN LEIB-"
36950 PRINT"   EIGENER BEI CLEOPATRA"
37000 PRINT"   WAERE JA AUCH NICHT VON"
37050 PRINT"   DER HAND ZU WEISEN."
37100 FORT=1TO4000:NEXTT:GOTO39000
38000 GOSUB54000
38050 POKE53280,1
38100 PRINT""
38150 PRINT"   CAESAR BEGLUECKWUENSCHT DEN SIEGER"
38200 PRINT:PRINTTAB(16)G$
38250 PRINT:GOSUB54000
38300 PRINT"   EINE HERVORRAGENDE STRATEGISCHE"
38350 PRINT"   LEISTUNG. JEFFERSON, KARL DER"
38400 PRINT"   GROSSE UND ICH ZUSAMMEN"
38450 PRINT"   HAETTEN ES NICHT BESSER MACHEN"
38500 PRINT"   KOENNEN. DU WIRST AB HEUTE IN"
38550 PRINT"   MEINEN ENGSTEN FUEHRUNGSSTAB"
38600 PRINT"   UEBERNOMMEN UND BEKOMMST DEN"
38650 PRINT"   HONORIGEN AUFTRAG DIESE MERK-"
38700 PRINT"   WUERDIGEN GALLIER AUS DIESEM"
38750 PRINT"   DORF ZU VERTREIBEN.":GOSUB54000
38800 FORT=1TO4000:NEXTT:GOTO39000
39000 PRINT"    NOCH EINMAL?"
39100 GETNE$:IFNE$=""THEN39100
39200 IFNE$="J"THENCLR:GOTO100
39300 PRINT"":POKE53280,1:POKE53281,1
39400 PRINT"BIS"
39500 PRINT"BALD"
39600 PRINT"BEI"
39700 PRINT"C A E S A R"
39800 PRINT"":END
40000 PRINT""
40100 POKE53280,11:POKE53281,11
40200 PRINT"HALLO, HIER IST CAESAR. ICH HOERTE, DAS"
40300 PRINT"IHR STRATEGISCH WAS DRAUF HABT - SOSO -"
40400 PRINT"NA, DAS WERDEN WIR JA GLEICH SEHEN. ICH"
40450 PRINT"HABE MIR FUER EUCH EIN KLEINES SPIEL EIN";
40500 PRINT"-FALLEN LASSEN. MEIN GEDACHTES SCHLACHT-";
40600 PRINT"FELD HAT EINE GROESSE VON ACHTZEHN  MAL"
40700 PRINT"ELF FELDERN UND JEDER HEERESFUEHRER HAT"
40800 PRINT"FUENFUNDZWANZIG LEGIONAERE.JEDER HEERES-";
40900 PRINT"FUEHRER STELLT  SEINE EIGENE FORMATION "
41000 PRINT"AUF. DAHER SOLLTE DER GEGNER BEI DER JE-";
41100 PRINT"WEILIGEN EINGABE NICHT AUF DEN SCHIRM"
41200 PRINT"SCHAUEN.  EURE LEGIONAERE SPRINGEN ODER"
41300 PRINT"ZIEHEN  GERADE  ODER  SCHRAEG  IN  ALLE"
41400 PRINT"RICHTUNGEN. MIT EINEM LEGIONAER KANN SO"
41500 PRINT"WEIT GESPRUNGEN WERDEN,  WIE ES DIE JE-"
41600 PRINT"WEILIGE STELLUNG ERLAUBT. BEIM SPRINGEN"
41700 PRINT"BLEIBEN EIGENE LEGIONAERE STEHEN,  GEG-"
41800 PRINT"NERISCHE WERDEN GETOETET. ZIEL IST ES"
41900 PRINT"DIE ZWEI LAGERFELDER ( XX ) DES GEGNERS"
42000 PRINT"MIT EIGENEN LEGIONAEREN ZU BESETZEN."
42060 PRINT"      BITTE EINE TASTE DRUECKEN."
42100 GETB$:IFB$=""THEN42100
42150 GOSUB30000
42200 PRINT"CAESAR WUENSCHT NOCH DIE NAMEN DER ZWEI"
42300 PRINT"HEERESFUEHRER KENNENZULERNEN."
42400 PRINT""
42500 INPUT"1. HEERESFUEHRER";S1$
42600 IFLEN(S1$)>9THENPRINT"BEIM ZEUS, DAS IST ZU LANG":GOTO42500
42700 PRINT:INPUT"2. HEERESFUEHRER";S2$
42800 IFLEN(S2$)>9THENPRINT"BEIM ZEUS, DAS IST ZU LANG":GOTO42700
42900 W=INT(RND(1)*2)+1
43000 IFW=1THENSN$(1)=S1$:SN$(2)=S2$
43100 IFW=2THENSN$(1)=S2$:SN$(2)=S1$
43150 PRINT"  TON          ?"
```

Listing des Taktik- und Strategiespiels »Caesar« (Fortsetzung)

```
43170 GETSD$:IFSD$=""THEN43170
43190 IFSD$<>"J"THENSD=0
43200 FORT=1TO200:NEXTT
43300 PRINT"▓▓"SN$(1)" FAENGT AN."
43400 PRINT"▓▓▓▓DIE GOETTER MOEGEN EUCH BEISTEHEN."
43500 GOSUB9700
43600 PRINT"J"
43700 RETURN
50000 IFSD=0THENRETURN
50050 FORT=1TO5:READMX:READMY:READMD:NEXTT
50100 POKEL,15:POKETH,13:POKETL,15
50200 POKEA,3*16+15:POKEH,9
50300 READMX:READMY:READMD
50400 IFMX=-1THENRESTORE:RETURN
50500 POKEFH,MX:POKEFL,MY:POKEW9,65
50600 FORT=1TOMD:NEXTT
50700 POKEW9,0:GOTO50300
51000 POKEL,15:POKETH,13:POKETL,15
51100 POKEA,3*16+15:POKEH,9
51200 READMX:READMY:READMD
51300 IFMX=-1THENRESTORE:RETURN
51400 POKEFH,MX:POKEFL,MY:POKEW9,65
51500 FORT=1TOMD:NEXTT
51600 POKEW9,0:GOTO51200
52000 IFSD=0THENRETURN
52050 FORT=1TO10:READMX:READMY:READMD:NEXTT
52100 POKEL,15:POKETH,13:POKETL,15
52200 POKEA,3*16+15:POKEH,9
52300 READMX:READMY:READMD
52400 IFMX=-1THENRESTORE:RETURN
52500 POKEFH,MX:POKEFL,MY:POKEW9,65
52600 FORT=1TOMD:NEXTT
52700 POKEW9,0:GOTO52300
53000 IFSD=0THEN53700
53050 FORT=1TO15:READMX:READMY:READMD:NEXTT
53100 POKEA,3*16+15:POKEH,9
53200 READMX:READMY:READMD
53300 IFMX=-1THENRESTORE:GOTO53700
53400 POKEFH,MX:POKEFL,MY:POKEW9,33
53500 FORT=1TOMD:NEXTT
53600 POKEW9,0:GOTO53200
53700 IFI=1THEND(2)=D(2)-1:RETURN
53800 IFI=2THEND(1)=D(1)-1:RETURN
54000 IFSD=0THENRETURN
54050 FORT=1TO20:READMX:READMY:READMD:NEXTT:POKEL,10
54100 POKEA,3*16+15:POKEH,9
54200 READMX:READMY:READMD
54300 IFMX=-1THENRESTORE:RETURN
54400 POKEFH,MX:POKEFL,MY:POKEW9,33
54500 FORT=1TOMD:NEXTT
54600 POKEW9,0:GOTO54200
55000 IFSD=0THENRETURN
55100 POKEL,15:POKEA,16+9:POKEH,64
55200 POKEFH,29:POKEFL,69
55300 POKEW9,129
55400 FORT=1TO200:NEXTT
55600 POKEW9,0:POKEA,0:POKEH,0
55700 RETURN
56000 IFSD=0THENFORT=1TO60:NEXTT:RETURN
56100 IFKJ=0THENKJ=1:GOTO56300
56200 IFKJ=1THENKJ=0:GOTO56600
56300 POKEFH,20:POKEFL,170:POKEW9,17
56400 FORT=1TO25:NEXTT
56500 POKEW9,0:RETURN
56600 POKEFH,12:POKEFL,70:POKEW9,17
56700 FORT=1TO25:NEXTT
56800 POKEW9,0:RETURN
```

Listing des Taktik- und Strategiespiels »Caesar« (Schluß)



Ab geht die Post. Bei der Spielstärke 1 muß man sehr viel Fingerfertigkeit besitzen, um über die Runden zu kommen

Renn
ohne St

Wie auf dem Hockenheimring fühlt man sich bei dem Spielprogramm »Auto« für die Grundversion des VC 20. Es dokumentiert wieder einmal, daß man auch reine Basicprogramme sehr schnell machen kann.

Das ausgesprochene Spielprogramm »Auto« läuft nur auf der Grundversion des VC 20. Außer seiner guten Grafik (mit selbstdefiniertem Zeichensatz) und realistischem Sound ist das Spiel sehr schnell, obwohl es vollständig in Basic geschrieben ist. Dies liegt daran, daß der Hauptprogrammteil sehr knapp bemessen ist (Tastaturabfrage, (Errechnung der nächsten Bildschirmposition, Ton).

Am Anfang des Spieles kann man eine Spielstärke von 1 bis 9 eingeben, wobei »1« sehr schnell und »9« sehr langsam ist.

Ziel des Spieles ist es, auf der Rennstrecke möglichst

```
1 REM BY ANDREAS LERCH
10 POKE52,28:POKE56,28:CLR
20 PRINT"{CTRL×3}"CHR$(8);:POKE36879,25
30 FORI=1TO21
40 POKE646,RND(1)*6+2:PRINT" ";
50 NEXT
60 PRINT"{CTRL×12}                 "
70 PRINT"{CTRL×8}   VC-20 "
80 PRINT"{CTRL×8}         "
90 PRINT"{CTRL×8}   AUTO  "
100 PRINT"{CTRL×8}          "
110 PRINT"{CTRL×3}"CHR$(8);:POKE3679,25
120 FORI=1TO21
130 POKE646,RND(1)*6+2:PRINT" ";
140 NEXT
150 POKE36878,13:FORT=255TO128STEP-1:POKE36875,T:NEXTT:POKE36875,0
160 PRINT"{CTRL×8}SPIELSTAERKE?"
170 PRINT"{CTRL×10}(1-9)"
180 GETA$:IFA$=""THEN180
190 N=(VAL(A$)-1)*10
200 POKE36879,25:PRINT"{CTRL×14}****BITTE WARTEN****"
210 PL%=7168:H%=7176
220 RESTORE
230 FORI=0TO511:POKEPL%+I,PEEK(32768+I):NEXT
240 FORY=1TO5
250 FORI=0TO7:READA:POKEH%+I,A:NEXTI
260 H%=H%+8:NEXTY
270 DATA0,204,204,254,205,254,204,204
280 DATA8,20,127,127,20,20,127,127
290 DATA51,51,127,179,127,51,51,0
300 DATA254,254,40,40,254,254,40,16
310 DATA63,33,127,191,140,156,172,76
320 POKE36869,255
330 E=1:W=5:Z=6
340 S=8112
350 POKE36879,25
360 PRINT"{CTRL×2}┌────────────┐          .";
370 PRINT"{CTRL}|            |"
380 PRINT"{CTRL}|            |"
390 PRINT"{CTRL}|   ┌─────┐  └───────────┐";
400 PRINT"|   |     |              |";
410 PRINT"|   |     |              |";
420 PRINT"|   |     └─────────┐    |";
430 PRINT"|   |               |    |";
440 PRINT"|   |               |    |";
450 PRINT"|   └─────────────┐ |    |";
460 PRINT"|                 | ||   |";
470 PRINT"|                 | ||   |";
480 PRINT"|   ┌─────────────┘ ||   |";
490 PRINT"|   |           |   ||   |";
500 PRINT"|   |     ┌─────┘   ||   |";
510 PRINT"|   └─────┘         ||   |";
520 PRINT"|                   ||   |";
530 PRINT"|          ┌────────┘|   |";
540 PRINT"└────────┐ └─────────┘   |";
550 PRINT"         |       I       |";
560 PRINT"         |       I       |";
570 PRINT"         └───────────────┘";
580 POKE7856+7,5:POKE7856+9,5:POKE7856+11,5
590 E=1
600 S=8112
610 REM
```

ahrer
rzhelm

G,I,T	Zählvariablen für Sound- und Grafikbefehle
A$	Variable für Tastaturabfrage
Y,N	Variablen für Erzeugung der verschiedenen Spielstärken
PL%,H%,Y	Variablen für Erzeugung des neuen Zeichensatzes
W	Variablen für Anzahl der Wagen
R	Variable für Anzahl der Runden
S	Variable für Bildschirmposition des Wagens
M(1),M(2),(M3),M(4)	Variablen für die Anzahl der Bildschirmposition, die der Richtung entspricht
D	Variable zur Unterscheidung, ob es der erste oder ein anderer Spieldurchlauf ist
E	Laufvariable für M(1) bis M(4)
Z	Variable für Benzininhalt
A	Datavariable

Variablendefinition zu »Auto«

```
620 PRINT"▒"
630 PRINT"▒WAGEN  :"W
640 PRINT"▒RUNDEN:"R
650 IFD=1THENRETURN
660 POKES,E
670 M(1)=1:M(2)=-22:M(3)=-1:M(4)=22
680 M(E)=1
690 GETA$
700 FORY=1TON:NEXTY
710 IFA$="L"THENGOSUB790
720 IFA$="R"THENGOSUB810
730 IFPEEK(S+M(E))=160THENGOTO750
740 GOTO830
750 S=S+M(E)
760 POKES,E:POKES-M(E),160
770 POKE36878,10:POKE36876,200:POKE36876,0
780 GOTO690
790 E=E+1:IFE=5THENE=1
800 RETURN
810 E=E-1:IFE=0THENE=4
820 RETURN
830 IFPEEK(S+M(E))<>137THEN870
840 POKE36878,14:FORI=1TO2:POKE36876,235:FORT=1TO9:NEXT:POKE36876,231:FORT=1TO9
:NEXTT,I
850 R=R+1:POKE36878,0:POKE36876,0:GOSUB1130:Z=Z-1:IFZ=0THENGOTO940
860 GOTO930
870 IFS+M(E)=7910ORS+M(E)=7932THENGOTO1230
880 GOSUB910
890 POKE36878,13:FORT=150TO220
900 POKE36877,T:FORG=1TO5:NEXTG,T:POKE36877,0:POKE36878,0:GOTO920
910 POKES,32+128:            POKES+M(E),73:RETURN
920 W=W-1:IFW=0THENGOTO940
930 D=0:POKE198,0:GOTO350
940 PRINT"▒":POKE36878,14:FORT=128TO255:POKE36876,T:NEXT:POKE36876,0:
950 POKE36869,240
960 IFPEEK(828)<RTHENPOKE828,R
970 POKE36879,8
980 PRINT"▒"CHR$(8);:POKE3679,8
990 FORI=1TO21
1000 POKE646,RND(1)*6+2:PRINT" ";
1010 NEXT
1020 PRINT"▒DEIN SCORE:"R
1030 PRINT"▒HIGHSCORE:";PEEK(828)
1040 PRINT"▒"CHR$(8);:POKE3679,8
1050 FORI=1TO21
1060 POKE646,RND(1)*6+2:PRINT" ";
1070 NEXT
1080 PRINT"▒NOCH EIN SPIEL?"
1090 PRINT"▒(J/N)"
1100 GETA$:IFA$="J"THEND=0:R=0:PRINT"▒":SC=0:GOTO200
1110 IFA$="N"THENEND
1120 GOTO1100
1130 IFR/5<>INT(R/5)THENRETURN
1140 POKE36878,13
1150 FORI=1TO4
1160 F=195:GOSUB1180:F=207:GOSUB1180:F=251:GOSUB1180:F=225:GOTO1200
1180 POKE36876,F:GOSUB1220:RETURN
1200 NEXTI
1210 POKE36876,0:POKE36878,0:RETURN
1220 FORY=1TO20:NEXTY:RETURN
1230 Z=6:FORT=1TO4500:NEXT:POKES,96+128:S=S+2:POKES,E:POKES-1,221:GOTO690
```

Listings des Basicprogramms »Auto«

lange zu fahren. Dazu hat man am Anfang fünf Wagen zur Verfügung.

Man verliert einen Wagen, wenn man an die Abgrenzung der Strecke kommt oder über fünf Runden nicht zur Tankstelle gefahren ist.

Im Eifer des Gefechts das Tanken nicht vergessen

Als kleine Hilfe wird man jede fünfte Runde akustisch in Kenntnis gesetzt, daß man tanken muß. Zur Tankstelle kommt man, wenn man in die erste Einbiegung auf der linken Seite fährt (siehe Bild).

Zwei Tasten sind genug

Gesteuert wird der Wagen über 'R' wie rechts und 'L' wie links.

In der Mitte des Bildschirms sieht man, wieviele Wagen man noch hat und wieviele Runden man gefahren ist.

Am Ende des Spiels wird der eigene Score und der Highscore angezeigt, welcher im Kassettenpuffer abgelegt wird. Es erfolgt die Frage nach einem weiteren Spiel, die mit »J« (Ja) oder »N« (Nein) zu beantworten ist.

Viel Spaß beim Spielen wünscht

(Andreas Lerch)

Erste Hilf

Wem wäre das noch nicht passiert: NEW. Nur drei Buchstaben, aber die Arbeit von Stunden oder gar Tagen ist verloren, wenn das Programm noch nicht gespeichert war. Da hilft dann nur noch eines: das VC 20-Programm »Erste Hilfe«. Laden, starten — und schon ist das gelöschte Programm wieder da. Wie neu!

Das grenzt nicht etwa an Zauberei, sondern ist möglich, weil ein Basic-Programm im Speicher des VC 20 mit dem Befehl NEW gar nicht gelöscht wird. Obwohl es nicht den Anschein hat, es ist weiterhin vorhanden. NEW setzt lediglich alle Zeiger zurück und schreibt in die beiden ersten Adressen je eine Null. Will man dem Speicher das nun im Verborgenen liegende Programm wieder entlocken, müssen diese Parameter rekonstruiert werden.

Da wären zunächst die beiden Nullen am Beginn. Sie besagen nichts anderes, als daß sich hier das Programmende befindet. Bei noch vorhandenem Programm enthalten diese Zellen die Koppeladresse zur zweiten Zeile, das heißt, die Speicheradresse, an der diese beginnt. Um sie zu finden, muß in der ersten Programmzeile nach der Null gesucht werden, die das Zeilenende markiert. Plus 1 und die Koppeladresse, zerlegt in Low- und Highbyte, kann wieder am angestammten Platz abgelegt werden. Der zweite Schritt besteht darin, das Programmende wiederzufinden. Mit Hilfe jener Koppeladressen am Beginn jeder Zeile ist auch das relativ einfach. Die erste verrät, wo die zweite steht, diese zeigt auf die dritte und so weiter. Das Programmende ist erreicht, wenn das Highbyte der Koppeladresse eine Null ist. Eine Speicherstelle weiter beginnt dann der Sektor, in dem die Variablen abgelegt werden. Diese Adresse wird dem Computer in den Speicherstellen 45 und 46 mitgeteilt. Der Befehl CLR paßt dann alle übrigen Parameter diesem Wert an. Damit ist das gelöschte Programm wieder da und kann gelistet, bearbeitet, gestartet oder abgespeichert werden.

Da eine solche Rekonstruktion von Hand ein recht mühsames Unterfangen ist, nimmt »Erste Hilfe« dem Anwender diese Arbeit ab. Nicht allein aus Gründen der Geschwindigkeit handelt es sich dabei um ein Maschinenprogramm. Vor allem muß es so in den Speicher gebracht werden, daß es das zu rettende Programm nicht überschreibt und dadurch vollends zerstört. Es ist daher in einem

```
0 REM HELP SYS 678
10 PRINTCHR$(147)CHR$(31)
20 FORA=678TO755:READB
30 POKEA,B:X=X+B:NEXT
40 IFX><10962THENF$="IN DATAS!":GOTO90
50 A=PEEK(43)+256*PEEK(44)+4
60 IFPEEK(A)><143GOTO80
70 A=A+2:IFPEEK(A)=72GOTO100
80 F$="IM KOPF!"
90 PRINT"FEHLER "F$:END
100 PRINT"ALLES OK!":PRINT
110 POKE186,1:REM*** KASSETTE ***
120 POKE187,AAND255:POKE188,A/256
130 POKE193,166:POKE194,2
140 POKE174,244:POKE175,2
150 POKE183,12:POKE185,1
500 POKE157,128:SYS63106

READY.
```

Listing 2. Basic-Lader für die Kassettenversion

```
02A6  A5 2B     LDA $2B
02A8  18        CLC          ;Basic-Programmstart Lowbyte
02A9  69 04     ADC #$04
02AB  85 FD     STA $FD      ;Zeiger auf Start + 4 setzen
02AD  A5 2C     LDA $2C      ;Highbyte
02AF  69 00     ADC #$00     ;Carry addieren
02B1  85 FE     STA $FE
02B3  A0 00     LDY #$00     ;Zaehler initialisieren
02B5  B1 FD     LDA ($FD),Y  ;Byte holen
02B7  F0 08     BEQ $02C1    ;Null = Zeilenende?
02B9  C8        INY
02BA  C0 58     CPY #$58     ;88 Bytes geprueft?
02BC  D0 F7     BNE $02B5
02BE  4C 08 CF  JMP $CF08    ;Aussprung mit 'SYNTAX ERROR' (SYS 53000)
02C1  C8        INY
02C2  98        TYA
02C3  A0 00     LDY #$00
02C5  18        CLC
02C6  65 FD     ADC $FD      ;Zaehler + 1 zur Startadresse
02C8  91 2B     STA ($2B),Y  ;neue Koppeladresse setzen
02CA  85 FD     STA $FD      ;Zeiger auf 2. Programmzeile
02CC  90 02     BCC $02D0    ;Carry?
02CE  E6 FE     INC $FE      ;Highbyte korrigieren
02D0  A5 FE     LDA $FE
02D2  C8        INY
02D3  91 2B     STA ($2B),Y  ;Koppeladresse high
02D5  88        DEY          ;Zaehler auf Null
02D6  B1 FD     LDA ($FD),Y  ;Koppeladresse zur naechsten Zeile holen
02D8  AA        TAX          ;und retten
02D9  C8        INY
02DA  B1 FD     LDA ($FD),Y  ;Adresse high holen
02DC  F0 07     BEQ $02E5    ;Null = Programmende?
02DE  85 FE     STA $FE      ;Zeiger auf naechste Zeile setzen
02E0  86 FD     STX $FD
02E2  4C D5 02  JMP $02D5    ;und weitermachen
02E5  A5 FD     LDA $FD
02E7  18        CLC
02E8  69 02     ADC #$02
02EA  85 2D     STA $2D      ;Endadresse + 1 = Variablenbeginn
02EC  A5 FE     LDA $FE      ;Highbyte holen
02EE  20 55 C6  JSR $C655    ;Teil des NEW-Befehls (SYS 50754):
                             ;Carry addieren
                             ;Highbyte Variablenspeicher setzen
                             ;CHARGET-Routine auf Programmbeginn setzen
                             ;und Variablen loeschen (CLR: SYS 50782)
02F1  4C 9C C6  JMP $C69C    ;Aussprung mit LIST (SYS 50844)
```

Listing 1. Assemblerprogramm »Erste Hilfe«

```
10 PRINTCHR$(147)CHR$(31)
20 FORA=678TO755:READB
30 POKEA,B:X=X+B
40 NEXT:IFX=10962GOTO90
50 PRINT"FEHLER IN DATAS!"
60 END
90 PRINT"ALLES OK!"
100 OPEN1,8,1,"HELP SYS 678"
110 PRINT#1,CHR$(166)CHR$(2);
200 FORA=678TO755:B=PEEK(A)
210 PRINT#1,CHR$(B);
250 NEXT:CLOSE1

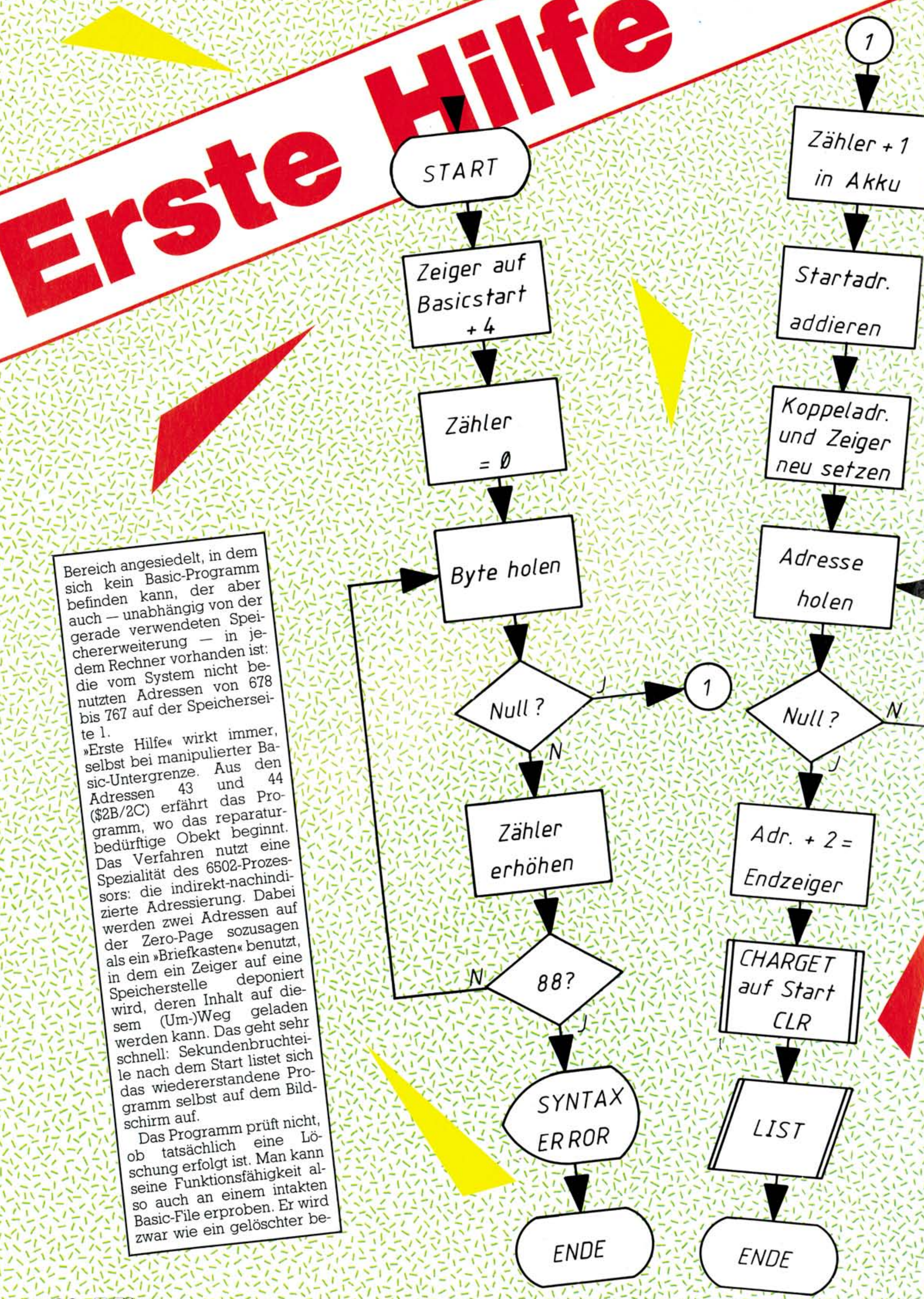
READY.
```

Listing 3. Basic-Lader für die Diskettenversion

```
1000 DATA165,43,24,105,4
1001 DATA133,253,165,44,105
1002 DATA0,133,254,160,0
1003 DATA177,253,240,8,200
1004 DATA192,88,208,247,76
1005 DATA8,207,200,152,160
1006 DATA0,24,101,253,145
1007 DATA43,133,253,144,2
1008 DATA230,254,165,254,200
1009 DATA145,43,136,177,253
1010 DATA170,200,177,253,240
1011 DATA7,133,254,134,253
1012 DATA76,213,2,165,253
1013 DATA24,105,2,133,45
1014 DATA165,254,32,85,198
1015 DATA76,156,198

READY.
```

Listing 4. Die gemeinsamen DATA-Werte für die beiden Lader



Bereich angesiedelt, in dem sich kein Basic-Programm befinden kann, der aber auch — unabhängig von der gerade verwendeten Speichererweiterung — in jedem Rechner vorhanden ist: die vom System nicht benutzten Adressen von 678 bis 767 auf der Speicherseite 1.

»Erste Hilfe« wirkt immer, selbst bei manipulierter Basic-Untergrenze. Aus den Adressen 43 und 44 ($2B/2C) erfährt das Programm, wo das reparaturbedürftige Obekt beginnt. Das Verfahren nutzt eine Spezialität des 6502-Prozessors: die indirekt-nachindizierte Adressierung. Dabei werden zwei Adressen auf der Zero-Page sozusagen als ein »Briefkasten« benutzt, in dem ein Zeiger auf eine Speicherstelle deponiert wird, deren Inhalt auf diesem (Um-)Weg geladen werden kann. Das geht sehr schnell: Sekundenbruchteile nach dem Start listet sich das wiedererstandene Programm selbst auf dem Bildschirm auf.

Das Programm prüft nicht, ob tatsächlich eine Löschung erfolgt ist. Man kann seine Funktionsfähigkeit also auch an einem intakten Basic-File erproben. Er wird zwar wie ein gelöschter be-

Flußdiagramm von »Erste Hilfe«

handelt, aber nicht verändert, da alle Werte so rekonstruiert werden, wie sie im Normalfall auch vorhanden sind. Sollte das aufgelistete Ergebnis Abweichungen vom Urzustand aufweisen, dann sind diese auf irreparable Zerstörungen zurückzuführen. Nach dem versehentlichen Löschen sollten keine Manipulationen mehr vorgenommen werden. Ein LIST-Versuch schadet nicht, aber jede von nun an verwendete Variable überschreibt den ungeschützt im Speicher liegenden File. »Erste Hilfe« kann natürlich nicht erkennen, wenn solche Veränderungen bereits eingetreten sind. Ein Aussprung unter Anzeige eines »Syntax Errors« erfolgt nur, wenn in der ersten Programmzeile nach der maximal zulässigen Zahl von 88 Bytes noch keine End-Null vorgefunden wurde. So kann es vorkommen, daß sich der Computer in einem weitgehend zerstörten File verirrt und festläuft. Der Verlust durch das notwendige Abschalten ist dann kein allzu großer, denn ein solches Programm wäre ohnehin nicht mehr zu retten gewesen.

Zeiger setzen

Störungen des Programms »Erste Hilfe« sind bisher nur vom Commodore-Modul »Super-Erweiterung« (VC1211A) bekannt. Vor dem Laden des Maschinenprogramms muß dieses Modul durch den im Direktmodus eingegebenen Befehl SYS 64818 abgeschaltet werden, wobei zu beachten ist, daß dadurch ein eventuell heraufgesetzter Basic-Start normalisiert wird und gegebenenfalls wieder angepaßt werden muß, bevor »Erste Hilfe« zur Anwendung kommt.

Assembler-Programmierer können das Maschinenprogramm nach dem dokumentierten Listing 1 eingeben und vom Monitor aus abspeichern. Für Basic-Programmierer stehen zwei verschiedene Versionen zur Auswahl. Das als »Kassettenversion« bezeichnete Listing 2 kann auch für die Diskette benutzt werden, wenn die Zeile 110 abgeändert wird in POKE 186,8. Hier wird die Gerätenummer hinterlegt. Die Adressen 187 und 188 enthalten die Adresse des Namens, unter dem das Programm abgespeichert werden soll. Dieser Name ist in der REM-Zeile mit der Nummer 0 abgelegt. Sie ist deshalb unbedingt erforderlich, weil gezielt danach gesucht wird. Die Speicherstelle 183 weist die Länge des Filenamens aus, 185 die Sekundäradresse 1, die dafür sorgt, daß das Programm nicht wie gewöhnlich an den Basic-Start, sondern nach Adresse 678 geladen wird. Von dort aus wird auch abgespeichert: 193/194 enthalten die Startadresse, 174/175 die Endadresse plus 1.

Bei Verwendung einer Diskette kann man sich die vielen POKEs jedoch sparen und die Bytes in Form von ASCII-Codes direkt auf die Floppy schreiben (siehe Listing 3). Dieses Verfahren ist ausschließlich für die Diskette geeignet. Der DATA-Block (Listing 4) ist in beiden Fällen derselbe. Gespeichert werden sollte der File unter dem angegebenen Namen, der die Startadresse enthält, die dadurch nicht in Vergessenheit geraten kann. Für das Laden von Diskette ist LOAD "H*",8,1 einzugeben. Vorteilhafter ist hier das Laden von der Kassette, weil ein einfaches LOAD ohne weitere Angaben genügt. Die Sekundäradresse ist nicht erforderlich, da das Betriebssystem des VC 20 im Bandheader das absolut zu ladende Maschinenprogramm erkennt. Mit SYS 6789 wird dann umgehend »Erste Hilfe« geleistet.

(Helmut Welke)

Wanted

Der Commodore 64 und der VC 20 sind die absoluten Marktführer bei den Heimcomputern. Sie — unsere Leser — haben mit diesem leistungsfähigen System Ihre ersten, zweiten und n-ten Erfahrungen gesammelt, waren begeistert von den vielfältigen Möglichkeiten, die mit diesen beiden Computern geboten wurden, oder aber auch gelegentlich enttäuscht von deren Unzulänglichkeiten (keine Reset- oder Escape-Taste, keine deutsche Tastatur, mangelnde Dokumentation und vieles andere mehr). Einsteiger in die Computerei hatten und haben ihre Probleme mit dem Commodore 64 und dem VC 20. Profis, Semi-Profis und solche, die es werden wollen, könnten bei der Bewältigung dieser Anfangsschwierigkeiten behilflich sein. Viele nützliche Routinen die den Umgang mit den Commodores erleichtern, liegen in den Schubladen, und wurden nicht veröffentlicht. Senden Sie uns Ihre Tips & Tricks, Utilities, Anwendungsprogramme und Spiele. Viele wären dankbar für eine Trace-Routine, einen deutschen Zeichensatz, eine einfache Tabellenkalkulation, eine interessante Anwendung oder für ein spannendes Spiel zum Entspannen nach harter Programmierarbeit. Einige werden vielleicht einwenden: Mich interessiert kein Renumberprogramm, ich habe Simons Basic oder Exbasic Level II, der soll sich das doch kaufen. Nun jeder ist nicht in der glücklichen Lage eines wohlgefüllten Geldbeutels. Also ran an den Commodore und die Tips & Tricks, Anwendungsprogramme oder Spiele eingesandt. Worauf man bei der Einsendung eines Programms zu achten hat, wird auf Seite 131 erläutert.

Die 64'er-Redaktion ist aber nicht nur an Programmen oder Tips & Tricks interessiert. Wir suchen auch Leute, die sich auf einem bestimmten Gebiet besonders gut auskennen. Es kann sich dabei um das Betriebssystem handeln oder sich um die Grafik drehen, mit Programmiersprachen (Basic, Pascal, Forth, Logo...) zu tun haben, die Hardware betreffen (selbstgebastelte Erweiterungen, gekaufte Expansions etc.) oder um Themen wie Beschaffung von Software, die neuesten Spiele und vieles andere mehr gehen. Schreiben Sie uns einfach, welche Vorschläge Sie haben oder senden Sie gleich einen fertigen Artikel ein. Wollen Sie nicht Ihr Wissen (gegen ein angemessenes Honorar bei Veröffentlichung) anderen mitteilen?

Auch der Anfänger ist aufgerufen, seine Probleme nicht einfach unter den Tisch zu kehren. Nur wer fragt, bekommt eine Antwort. Das 64'er Magazin soll nicht nur ein Forum für die Freaks sein, sondern will auch dem Neuling im Umgang mit dem Computer Hilfestellung bieten. Setzen Sie sich mit der 64'er Redaktion (Hans-Pinsel-Str. 2, 8013 Haar b. München) in Verbindung.

Disk Copy

Wer hat sich als stolz Kopieren von Program

Schlimmer wird es, wenn es sich um Maschinenprogramme oder gar um sequentielle Files handelt. Das Anlegen einer Sicherheitskopie wird zur zeitraubenden, umständlichen Prozedur und unterbleibt deshalb, bis man eines Tages feststellt, daß das Original aus irgendwelchen Gründen nicht mehr läuft. Mir ist das mit einer Datei passiert, die aus über 400 Einträgen bestand und die ich eines Tages einfach nicht mehr einladen konnte. Seitdem gibt es bei mir von allen wichtigen Disketten Sicherheitskopien, bei deren Erstellung mir das folgende Programm gute Dienste leistet.

Das Besondere an diesem Programm ist, daß alle Arten von Files mit Ausnahme relativer Dateien kopiert werden können und das recht komfortabel und schnell. Der Trick besteht darin, daß die Files, die man kopieren möchte, der Reihe nach in den Speicher gelesen werden bis dieser voll ist. Da das Programm auch die »versteckten« RAM-Bereiche mitbenutzt, die man normalerweise gar nicht ansprechen kann, weil Basic-Interpreter und Betriebssystem an derselben Speicheradresse liegen, kommt man in einem Durchgang auf stattliche 226 Blöcke (zirka 56 KByte). Sollte das noch nicht ausreichen, startet das Programm einen zweiten oder auch dritten Durchgang. Dann ist spätestens die gesamte Diskette kopiert (bei ganz ungünstigen Verhältnissen wird noch ein vierter Durchgang gebraucht). Damit ergeben sich Zeiten von unter 20 Minuten für eine Komplettkopie.

```
100 REM *** INITIALISIERUNG ***
110 POKE56,PEEK(46)+14:CLR:RB=255-PEEK(56):PA=1:AN=0:BL=0:NF$=""
120 PE=PEEK(45)+256*PEEK(46):MR=PE-135:MW=PE-79:MD=PE-24
130 DIMNF$(140),CF%(140),BL%(140),P%(10),AL%(90),AH%(90)
140 P%(0)=0:AL%(0)=0:AH%(0)=PEEK(56)-1
150 :
160 REM *** MENUE ***
170 PRINT""TAB(9)"***** DISK COPY *****":PRINTTAB(10)"VON D.WEINECK 2/84"
180 PRINT"1.  DIRECTORY
190 PRINT"2.  KOPIEREN
200 PRINT"3.  FORMATIEREN
210 PRINT"4.  ENDE
220 PRINTSPC(212)"BITTE WAEHLEN SIE
230 GETDC$:DC=VAL(DC$):IFDC<1ORDC>4THEN230
240 ONDCGOTO910,270,700,670
250 :
260 REM *** KOPIEREN ***
270 PRINT"ORIGINALDISKETTE EINLEGEN"
280 GOSUB990
290 REM *** FILES EINLESEN ***
300 OPEN1,8,0,"$0"
310 GOSUB760:IFNF$<>""THEN340
320 IFST=0THEN310
330 GOTO350
340 BL%(AN)=ASC(BL$+CHR$(0)):NF$(AN)=NF$:IFST=0THENAN=AN+1:NF$="":GOTO310
350 CLOSE1:AN=AN-1:IFAN=0THENPRINT"LEERE DISKETTE":GOSUB990:RUN
360 REM *** KOPIERAUSWAHL ***
370 PRINT"ANTWORTEN SIE MIT J/N"
380 FORI=1TOAN:PRINTBL%(I);TAB(5)NF$(I)" ? ";:POKE198,0
390 WAIT198,1:GETA$:IFA$="J"THENCF%(I)=-1:BL=BL+BL%(I):PRINTTAB(30)" JA ":GOTO
420
400 CF%(I)=0:IFA$<>"N"THEN390
410 PRINTTAB(30)"NEIN"
420 IFBL>RBTHENP%(PA)=I-1:PA=PA+1:BL=BL%(I)
430 NEXTI:P%(PA)=AN
440 IFBL=0THEN640
450 REM *** KOPIE ***
460 PRINT"KOPIE IN ARBEIT"
470 FORI=1TOPA
480 FORRW=0TO1:NR=0:IFRW=1THENPRINT"ZIELDISK EINLEGEN":GOSUB990
490 FORJ=P%(I-1)+1TOP%(I)
500 IFNOTCF%(J)THENNEXTJ:GOTO540
510 NF$=NF$(J):PRINTBL%(J);TAB(5)NF$:GOSUB570:IFST=0ORST=64THEN530
520 GOSUB880:RUN
530 NEXTJ
540 NEXTRW:IFI=PATHEN640
550 PRINT"ORIGINALDISK EINLEGEN":GOSUB990
```

Wie funktioniert's?

Sehen wir uns zuerst das Basicprogramm an (Bild 1): Ich habe es absichtlich in mehrere Teile zerlegt, um es übersichtlicher zu machen. Die REM-Zeilen können beim Eintippen natürlich ebenso wegfallen wie die Zeilen, in denen nur ein Doppelpunkt steht.

Zeile 100 bis 140

setzt zuerst die Speicherobergrenze für Basic herunter (Speicherstelle 56) und berechnet, wieviel Speicher zum Kopieren zur Verfügung steht (RB). Um spätere Erweiterungen einbauen zu können (zum Beispiel Backup für relative Dateien), arbeite ich hier nicht mit festen Zahlen, sondern mit Variablen, die das Programm selbst berechnet. Das gilt ebenso für die Startadressen der Maschinenroutinen. Diese befinden sich direkt hinter dem Basic-Teil und brauchen deshalb nicht erst über DATA-Zeilen bei jedem Programmlauf »eingepoked« werden. Das spart nicht nur Zeit, sondern vor allem auch Speicherplatz (zirka 700 Bytes). Dafür müssen Sie beim Abschreiben zwei Teile zusammenfügen. Im Initialisierungsteil werden auch alle Variablen und Arrays eingerichtet (siehe Variablentabelle).

Zeile 160 bis 240

enthält das Menü. Sie können hier jederzeit weitere Funktionen einfügen.

Wird der Programmteil »Formatieren« gewählt, springt das Programm in die Zeilen 700 bis 750. Interessant ist hier die Möglichkeit, bei der Frage nach der Disk — ID keine Eingabe zu machen, sondern »RETURN« zu drücken. Dies ist bei Disketten sinnvoll, die bereits formatiert sind, aber gelöscht werden sollen. Das DOS der 1541 löscht dann nur die BAM und die erste Seite des Directory, was viel schneller geht als vollständiges Neuformatieren. Das Programm schließt diesen Teil mit Ausgabe der Fehlermeldung ab und springt wieder ins Menü.

Der Programmteil »Directory« ermöglicht ein schnelles Einlesen des Directorys, natürlich ohne Programmverlust. Man kann sich so einen kurzen Überblick über Original- und Zieldiskette verschaffen, ohne die Kopierroutine aufrufen zu müssen. Hier wird ein Maschinenteilprogramm benutzt, um Speicherplatz und Zeit zu sparen.

Zeile 270 bis 650

Kommen wir nun zum Kern der Sache, dem eigentlichen Kopierteil. Dieser befindet sich in den Zeilen 270 bis 650. Im ersten Teil (Zeile 270 bis 350) werden alle Files, die auf der Diskette sind, in ein Stringarray (NF$(I)) eingelesen,

er Besitzer eines Commodore 64 und einer 1541-Floppy noch nicht beim
men geärgert? Solange es sich um reine Basic-Programme handelt, geht es noch: Originaldiskette einlegen, Programm laden, Diskette wechseln, Programm »saven«, Diskette wechseln, Programm laden....

```
560 NEXTI:RUN
570 IFRW=1THEN610
580 OPEN1,8,5,NF$+",R":POKE252,0:POKE253,AH%(NR)+1
590 SYSMR:NR=NR+1:AL%(NR)=PEEK(254):AH%(NR)=PEEK(255)
600 CLOSE1:RETURN
610 OPEN1,8,5,NF$+",W":POKE252,0:POKE253,AH%(NR)+1
620 POKE254,AL%(NR+1):POKE255,AH%(NR+1):SYSMW
630 NR=NR+1:CLOSE1:RETURN
640 PRINT"KOPIE FERTIG !
650 GOSUB990:RUN
660 REM *** ENDE ***
670 POKE56,160:END
680 :
690 REM *** FORMATIEREN ***
700 INPUT"DISKNAME";FO$:ID$="":INPUT"DISK-ID";ID$:IFID$<>""THENID$=","+ID$
710 FO$=FO$+ID$
720 PRINT"BITTE ZIELDISKETTE EINLEGEN"
730 GOSUB990
740 OPEN1,8,15,"N:"+FO$:CLOSE1
750 GOSUB880:GOTO170
760 REM DIRECTORY EINLESEN
770 GET#1,A$,B$
780 GET#1,BL$,B$
790 GET#1,A$
800 GET#1,B$:IFST<>0THENRETURN
810 IFB$<>CHR$(34)THEN800
820 GET#1,B$:IFB$<>CHR$(34)THENNF$=NF$+B$:GOTO820
830 GET#1,B$:IFB$=CHR$(32)THEN830
840 NF$=NF$+","+B$:FORI=0TO1:GET#1,B$:NF$=NF$+B$:NEXT
850 GET#1,B$:IFB$<>""THEN850
860 RETURN
870 REM *** FEHLER-AUSGABE ***
880 OPEN15,8,15:INPUT#15,A,B$,C,D:PRINTA;B$;C;D:CLOSE15:GOSUB990:RETURN
890 :
900 REM *** DIRECTORY ***
910 PRINT""
920 OPEN3,8,0,"$0":GET#3,A$,A$
930 GET#3,A$,A$,BL$,BH$
940 IFA$=""THENCLOSE3:GOTO980
950 BL$=BL$+CHR$(0):BH$=BH$+CHR$(0)
960 PRINT256*ASC(BH$)+ASC(BL$);
970 SYSMD:GOTO930
980 GOSUB 990:GOTO170
990 PRINTSPC(69)"________":PRINTSPC(29)"*TASTE*"
1000 POKE198,0:WAIT198,1:GETA$:RETURN
READY.
```

Bild 1. Basicprogramm »Disk Copy«

die zugehörigen Blocklängen in ein Variablenarray (BL%(I)). Das dauert etwas länger, weil der Speicherplatz begrenzt ist und der Basic-Interpreter mitunter eine Garbage-Collection (Müll-Sammlung) durchführt, um Platz zu schaffen. Die Anzahl der Files wird in der Variablen AN festgehalten. Ist die Diskette leer, weil man zum Beispiel noch die gerade formatierte Zieldiskette im Laufwerk hatte, springt das Programm ins Menü zurück.

In Zeile 370 bis 440 erfolgt die Auswahl, welche Files kopiert werden sollen. Das Programm schreibt dazu die einzelnen Namen mit der zugehörigen Blocklänge auf den Bildschirm und Sie können mit »J« oder »N« aussuchen. Die Entscheidung merkt sich das Programm wieder in einem Variablenarray (CF%(I)): Ist CF%=—1, wird kopiert, sonst nicht. Gleichzeitig wird in der Variablen BL aufaddiert, wieviele Blöcke zu kopieren sind, damit das Programm herausbekommt, ob ein Durchgang ausreicht oder nicht. Wenn Sie alle Files mit »N« kennzeichnen, springt das Programm wieder ins Menü.

Nachdem nun endlich alle Entscheidungen und Vorbereitungen abgeschlossen sind, geht es ans Kopieren (Zeile 460 bis 650). Der Reihe nach werden alle gekennzeichneten Files in den Speicher eingelesen. Dazu wird das Laufwerk mit »OPEN«-Befehlen angesprochen, weil so alle Arten von Files geladen und auch abgespeichert werden können. (Mit »LOAD« könnten nur Programmfiles geladen werden.) Erfreulicherweise enthält die Variable NF$ nicht nur den Namen des jeweiligen Files, sondern auch den Typ, also PRG, SEQ oder USR. Deshalb können alle Filetypen mit ein und derselben Routine verarbeitet werden.

Der Unterschied zwischen Lesen und Schreiben liegt lediglich darin, daß dem »OPEN«-Befehl im ersten Falle ein R (für Read), im zweiten ein W (für Write) angehängt wird. Die Datenübertragung selbst erledigt das Maschinenprogramm, dem wir uns gleich zuwenden werden. Um Variablen an diese Routinen übergeben zu können, benutzen wir die Zeropage-Speicherstellen 252 bis 255 ($FC bis $FF).

Werfen wir nun noch einen Blick auf die Maschinensprach-Teile (Bild 3 bis 5). Dabei soll vor allem erläutert werden, wie man den vollen RAM-Speicher des C 64 nutzen kann.

Dazu ist ein kurzer Blick auf die Speicheraufteilung des C 64 erforderlich, insbesondere den Teil ab $A000 bis $FFFF (dez. 40960 — 65535). Hier liegt normalerweise der Basic-Interpreter, der 8 KByte Adreßraum benötigt ($A000 - $C000). Darüber liegen in 4 KByte die Ein-Ausgabe-Einheiten ($D000 - $E000). Ganz oben im Speicher befindet sich das Betriebssystem, das genau wie der Basic-Interpreter 8 KByte belegt ($E000 - $FFFF). Zusätzlich ist aber der gesamte Bereich auch noch mit RAM bestückt. Woher weiß der Prozessor nun, was er benutzen soll? Lediglich 3 Bits in Speicherstelle 1 sind für die Auswahl zuständig: Bit 0 schaltet den Basic-Interpreter ein und aus, Bit 1 gleichzeitig Basic-Interpreter und Betriebssystem, Bit 2 ist für uns schon uninteres-

Variablen - Liste

RB:	Anzahl der zur Verfuegung stehenden RAM - Bloecke
PA:	Anzahl der Durchgaenge beim Kopieren
AN:	Anzahl der Files auf der Diskette
BL:	Anzahl der zu Kopierenden Bloecke
NF$:	Name und Typ des Files
PE:	Programmende
MR:	Adresse der Maschinenroutine zum Lesen
MW:	Adresse der Maschinenroutine zum Schreiben
MD:	Adresse der Maschinenroutine fuer Directory-Ausgabe
NF$(I):	Feld fuer Filenamen
CF%(I):	Feld fuer Kopierflags
BL%(I):	Blocklaenge der einzelnen Files
P%(I):	Anzahl der Files in einzelnen Durchgaengen
AL%(I):	Endadresse der einzelnen Files im Speicher (Low-Byte)
AH%(I):	Endadresse der einzelnen Files im Speicher (High-Byte)
RW:	Flag fuer Lesen oder Schreiben
I,J:	Schleifenvariable

Disk Copy Disk Copy Disk Copy

```
100 PRINT"BASIC - DATA - LADER FUER 'DISK COPY
110 INPUT"ANFANGSADRESSE";AD:ED=AD+135
120 FORI=ADTOED-1
130 READZ:POKEI,Z:NEXT
140 HB=INT(ED/256):LB=EDAND255
150 PRINT"LADEN SIE NUN DAS PROGRAMM 'DISK COPY'.
160 PRINT"GEBEN SIE NACH DEM LADEN EIN:":PRINT"POKE 45,"LB":POKE 46,"HB"
170 PRINT"DAS PROGRAMM BEFINDET SICH JETZT VOLL-
180 PRINT"STAENDIG IM SPEICHER UND KANN GE'SAVED' WERDEN.
190 DATA162,1,32,198,255,160,0,32,207,255,120,170,165,1,41,252,133,1,138,145
200 DATA252,165,1,9,3,133,1,88,32,183,255,201,64,240,11,201,0,208,7,200,208
210 DATA221,230,253,208,217,32,204,255,132,254,165,253,133,255,96,162,1,32
220 DATA201,255,160,0,120,165,1,41,252,133,1,177,252,170,165,1,9,3,133,1,88
230 DATA138,32,210,255,32,183,255,201,0,208,17,200,208,2,230,253,165,255,197
240 DATA253,208,217,196,254,144,213,240,211,76,204,255,162,3,32,198,255,32
250 DATA207,255,32,210,255,208,248,169,13,32,210,255,76,204,255,0,0,0
READY.
```

Bild 2. Basic Lader für »Disk Copy«

sant. Werden beide, Bit 0 und Bit 1, auf 0 gesetzt, ist zusätzlich auch noch der Ein-Ausgabebereich abgeschaltet. Eigentlich doch ganz einfach.

Der Teufel steckt wie fast immer im Detail: Wenn der Basic-Interpreter abgeschaltet ist, wie soll dann ein Basic-Programm laufen? Mehr noch, ohne sein Betriebssystem ist der Prozes-

Merge

Kleben per Software!

Wenn Sie ein Unterprogramm haben — zum Beispiel ein Formatierungsprogramm zum Erstellen von Tabellen — und sie möchten es an ein vorhandes Basicprogramm anhängen, ohne es extra eintippen zu müssen — mit Merge eine einfache Sache!

Sie müssen lediglich das Ende des Basic-Programms finden (3 Bytes 0) und die Adresse der zweiten Null in den Zeiger für Basic-Anfang einschreiben. Dann können Sie ein zweites Programm laden und modifizieren, ohne daß das erste Programm beeinflußt wird. Wenn sie anschließend wieder die ursprüngliche Startadresse in den Basic-Pointer schreiben, haben Sie ein einziges Programm.

Leider ist es eine langwierige Angelegenheit, das Ende eines Programms zu suchen, die gefundene Adresse in den Basic-Pointer einzupoken...... Das Programm Merge übernimmt diese Arbeit — und das Rücksetzen des Basic-Pointers auch.

Wenn Sie ein Programm geladen haben, so brauchen sie nur SYS 50000 einzugeben und das »Kuppelprogramm« Merge meldet sich mit

```
1 rem .........................
2 rem .                       .
3 rem .  merge 1.1     11/83  .
4 rem .                       .
5 rem .   von heinz boeffel   .
6 rem .    kantstrasse  12    .
7 rem .   6680 neunkirchen  7 .
8 rem .                       .
9 rem .........................
10 print chrS(14);chrS(147)
30 print"          ... merge 1.1 ..."
40 print"          von heinz boeffel";chrS(13);chrS(13)
50 print"Das Programm MERGE setzt den Zeiger"
60 print"fuer Basic-Anfang unmittelbar hinter"
70 print"das im Speicher befindliche Programm.";chrS(13)
80 print"Somit kann ein weiteres Programm hinter"
90 print"das bestehende geladen werden.";chrS(13)
100 print"Geben sie hierzu den Befehl SYS 50000"
110 print"ein. Das nachgeladene Programm kann"
120 print"genau wie das erstgeladene behandelt"
130 print"werden.";chrS(13)
150 print"Um den Basic-Zeiger wieder zurueck-"
160 print"zusetzen, geben Sie wieder SYS 50000"
170 print"ein. Die beiden Programme ergeben nun"
180 print"zusammen ein einziges Programm.";chrS(13)
190 print "Zum Laden bitte Leertaste druecken!"
200 get gS:if gS()" " then 200
210 print chrS(147);chrS(142)
220 for i=50000 to 50264:read q:poke i,q:next i
230 new
10000 data 169,255,133,2,165,43,201,1,208,13
10010 data 133,251,165,44,201,8,208,5,133,252
10020 data 76,125,195,165,251,133,43,165,252,133
10030 data 44,162,0,189,0,196,240,6,32,22
10040 data 231,232,208,245,96,160,0,177,43,208
10050 data 12,200,177,43,208,7,200,177,43,208
10060 data 2,133,2,230,43,208,2,230,44,165
10070 data 2,208,228,162,0,189,167,195,240,6
10080 data 32,22,231,232,208,245,96,13,13,32
10090 data 32,32,32,32,32,32,32,32,32,42
10100 data 42,42,32,77,69,82,71,69,32,49
10110 data 46,49,32,42,42,42,13,32,32,32
10120 data 32,32,32,32,32,32,32,86,79,78
10130 data 32,72,69,73,78,90,32,66,79,69
10140 data 70,70,69,76,13,13,32,32,32,32
10150 data 32,32,32,32,32,32,80,82,79,71
10160 data 82,65,77,77,32,79,78,32,72,79
10170 data 76,68,33,13,0,0,13,13,32,32
10180 data 32,32,32,32,32,32,32,32,42,42
10190 data 42,32,77,69,82,71,69,32,49,46
10200 data 49,32,42,42,42,13,32,32,32,32
10210 data 32,32,32,32,32,32,86,79,78,32
10220 data 72,69,73,78,90,32,66,79,69,70
10230 data 70,69,76,13,13,32,32,32,32,32
10240 data 32,32,32,32,32,80,82,79,71,82
10250 data 65,77,77,83,32,77,69,82,71,69
10260 data 68,33,13,0,0,0,0,0,0,0

ready.
```

Disk Copy Disk Copy Disk Copy

sor hilfloser als ein Blinder im Nebel.

Die Lösung liegt einfach darin, zu verhindern, daß der Prozessor überhaupt auf die Idee kommt, in sein Betriebssystem oder ins Basic hineinzuspringen. Letzteres ist nicht schwierig, denn wir befinden uns ja in einem Maschinenprogramm, wenn wir das Basic abschalten.

MERGE 1.1
VON HEINZ BOEFFEL
PROGRAMM ON HOLD!

Dann können Sie ein weiteres Programm (nur mit höheren Zeilennummern als das erste!) laden.

Das kann auch beispielsweise das Directory einer Diskette sein oder irgendwas, was Sie gerade ausprobieren möchten, sein. Mit NEW läßt sich dieses gleich wieder löschen.

Geben Sie dann wieder SYS 50000 ein und Merge meldet sich mit

MERGE 1.1
VON HEINZ BOEFFEL

PROGRAMMS MERGED!

Falls Sie das zweite Programm nicht durch NEW gelöscht haben, so ist aus den beiden Programmen ein einziges geworden. Haben Sie allerdings das zweite Programm vor dem Befehl SYS 50000 gelöscht, so erhalten Sie wieder den Ausgangszustand.

Merge ist vollständig in Maschinensprache geschrieben — daher beträgt die Ablaufzeit nur wenige Augenblicke.

Somit stellt Merge ein nützliches Hilfsmittel dar, das viele einzelne Eingaben (vor allem bei immer wieder verwendeten Unterprogrammen oder beim »Stricken« längerer Programme, die Sie zunächst abschnittweise abspeichern) erspart.

(Heinz Böffel)

Und bei der Bearbeitung eines solchen Programms kann nur dann etwas passieren, wenn der Prozessor das Programm verläßt. Das tut er allerdings jede 1/60 Sekunde, zum Beispiel um die interne Uhr weiterzustellen und nachzusehen, ob eine Taste gedrückt wurde etc. Das ist die sogenannte Interrupt-Routine. Wenn wir ihm die sperren, kann eigentlich gar nichts mehr schiefgehen. Und es funktioniert tatsächlich:

Im Leseteil des Maschinenprogramms wird zuerst der mit »OPEN« eröffnete Kanal als Eingabekanal gesetzt (FFC6). Dann wird ein Byte über diesen Kanal geholt. Jetzt sperrt der SEI (Set Interrupt) die Interruptroutine und wir können in aller Ruhe schalten und walten. Wir schieben das Byte ins X-Register, legen die beiden unteren Bits der Speicherstelle 1 auf 0, holen unser Byte aus dem X-Register und legen es im RAM ab. Jetzt setzen wir die Bits wieder auf 1, löschen die Interrupt-Sperre, und alles ist wieder in Ordnung. Nachdem unser Byte im RAM sicher untergebracht ist, fragen wir nun ab, ob vielleicht ein Fehler aufgetreten ist (FFB7) oder das Ende unseres Files erreicht ist. Wenn ja, springen wir ins Basic zurück und brechen im Fehlerfalle das Programm mit einer entsprechenden Meldung ab. Wenn nein, wenden wir uns dem nächsten Byte zu, das übertragen werden soll und behandeln es mit der gleichen Sorgfalt. Ganz zum Schluß müssen wir noch wieder die Kanäle zurücksetzen (Tastatur als Eingabe, Bildschirm als Ausgabe (FFCC)). Das alles klingt zwar umständlich und langwierig, geht aber in Wirklichkeit unglaublich schnell.

Das Schreiben auf Diskette bringt nichts grundsätzlich Neues, der ganze Vorgang läuft hier einfach andersherum ab. Unser Kanal 1 ist jetzt Ausgabekanal ($FFC9), und anstatt ein Byte von der Diskette zu holen, geben wir es aus ($FFD2).

Auch die Directory-Ausgabe folgt diesem Muster:

```
LESEN VON DISKETTE
,114D  A2 01     LDX #01
,114F  20 C6 FF  JSR FFC6
,1152  A0 00     LDY #00
,1154  20 CF FF  JSR FFCF
,1157  78        SEI
,1158  AA        TAX
,1159  A5 01     LDA   01
,115B  29 FC     AND #FC
,115D  85 01     STA   01
,115F  8A        TXA
,1160  91 FC     STA  (FC),Y
,1162  A5 01     LDA   01
,1164  09 03     ORA #03
,1166  85 01     STA   01
,1168  58        CLI
,1169  20 B7 FF  JSR FFB7
,116C  C9 40     CMP #40
,116E  F0 0B     BEQ 117B
,1170  C9 00     CMP #00
,1172  D0 07     BNE 117B
,1174  C8        INY
,1175  D0 DD     BNE 1154
,1177  E6 FD     INC   FD
,1179  D0 D9     BNE 1154
,117B  20 CC FF  JSR FFCC
,117E  84 FE     STY   FE
,1180  A5 FD     LDA   FD
,1182  85 FF     STA   FF
,1184  60        RTS
```

Bild 3. Für Interessierte: Maschinenroutine: »Lesen von Disk«

```
SCHREIBEN AUF DISKETTE
,1185  A2 01     LDX #01
,1187  20 C9 FF  JSR FFC9
,118A  A0 00     LDY #00
,118C  78        SEI
,118D  A5 01     LDA   01
,118F  29 FC     AND #FC
,1191  85 01     STA   01
,1193  B1 FC     LDA  (FC),Y
,1195  AA        TAX
,1196  A5 01     LDA   01
,1198  09 03     ORA #03
,119A  85 01     STA   01
,119C  58        CLI
,119D  8A        TXA
,119E  20 D2 FF  JSR FFD2
,11A1  20 B7 FF  JSR FFB7
,11A4  C9 00     CMP #00
,11A6  D0 11     BNE 11B9
,11A8  C8        INY
,11A9  D0 02     BNE 11AD
,11AB  E6 FD     INC   FD
,11AD  A5 FF     LDA   FF
,11AF  C5 FD     CMP   FD
,11B1  D0 D9     BNE 118C
,11B3  C4 FE     CPY   FE
,11B5  90 D5     BCC 118C
,11B7  F0 D3     BEQ 118C
,11B9  4C CC FF  JMP FFCC
```

Bild 4. In Assembler: Schreiben auf Diskette

```
DIRECTORY HOLEN
,11BC  A2 03     LDX #03
,11BE  20 C6 FF  JSR FFC6
,11C1  20 CF FF  JSR FFCF
,11C4  20 D2 FF  JSR FFD2
,11C7  D0 F8     BNE 11C1
,11C9  A9 0D     LDA #0D
,11CB  20 D2 FF  JSR FFD2
,11CE  4C CC FF  JMP FFCC
```

Bild 5. Laden des Directorys

Kanal 3 als Eingabe setzen ($FFC6), Zeichen für Zeichen holen ($FFCF) und — jetzt auf dem Bildschirm — ausgeben ($FFD2).

Wichtige Bedienungshinweise

So, nun steht dem Eintippen des Programms nichts mehr im Wege. Noch ein paar wichtige Hinweise: Die beiden Teile des Programms müssen beim ersten Mal zusammengefügt werden. Dazu gehen wir folgendermaßen vor:

1. Tippen Sie das Programm »Disk Copy« ab und speichern Sie es auf Diskette.
2. Starten Sie das Programm mit »RUN« und drücken Sie die »RUN/STOP«-Taste, wenn das Menü erscheint.
3. Geben sie ein: »PRINT PR« und schreiben Sie sich die angezeigte Zahl auf.
4. Tippen Sie das Programm »Basic-Data-Lader« ein und starten Sie es. Auf die Frage nach der Anfangsadresse geben Sie Ihre aufgeschriebene Zahl ein.
5. Folgen Sie genau den Anweisungen des Programs und geben Sie die beiden »POKE«-Befehle ein.
6. Speichern Sie das vollständige Programm auf Diskette ab.

Jetzt haben Sie das Programm gebrauchsfähig auf Diskette. Sie können auch beliebige Änderungen am Programm durchführen, der Maschinensprach-Teil wird sich immer automatisch mitverschieben.

Anpassung auf VC 20:

Das Programm läuft auch auf dem VC 20, für den ich es ursprünglich geschrieben hatte. Nur Zeile 110 muß geändert werden:

```
110 POKE56,PEEK(46)+14:
CLR:RB=PEEK(644)—PEEK
(56):...
```

Wenn Sie mit einer 1541-Floppy arbeiten, sollten Sie noch einfügen:

```
115 OPEN1,8,15,"UI-":
CLOSE1.
```

Und nun viel Spaß beim Kopieren.

(Dietrich Weineck)

Wollen Sie einen gebrauchten Computer verkaufen oder erwerben? Suchen Sie Zubehör? Haben Sie Software anzubieten oder suchen Sie Programme oder Verbindungen? Die FUNDGRUBE von 64'er bietet allen Computerfans die Gelegenheit, für nur DM 5,— eine private Kleinanzeige mit bis zu 5 Zeilen Text in der Rubrik Ihrer Wahl aufzugeben. Und so kommt Ihre private Kleinanzeige in die FUNDGRUBE der **Mai-Ausgabe** (erscheint am 19. April 84): Schicken Sie Ihren Anzeigentext bis zum 27. März 84 (Datum des Poststempels und Anzeigenschluß) an »64'er«. Später eingehende Aufträge werden in der **Juni-Ausgabe** (erscheint am 18. Mai 84) veröffentlicht.



COMMODORE VC 20

30 Modul Prog. auf Kassette + Porto für nur DM 40,—, pro NN. P. Kryger, Bülten 11, Postfach 1249, 3078 Stolzenau

Achtung! Superangebot! Achtung! Verkaufe gesamte VC 20 Software! Wert über 10000 DM! Über 200 Pgm. für 200 DM! Tel. 06196/22628. Achtung! Superangebot! Achtung!

Super-Aktion! Pgm's aus USA und England! Riesenauswahl und superbillig! Auch Tausch! Info 80 Pf. bei F. Langer, Hopfengarten 7, 6232 Bad Soden, Tel. 06196/22628

Steuerfuchs für VC 20 + 8 K 30 DM! Berechnung d. Erstattung/Verkäufe VC 20 2 8 K + Bücher + 50 Spiele f. 400 DM. J. Wiedelmann, Anhalter Str. 8, 4 Düsseldorf 1, Tel. 0211/214902

Hallo VC 20 Freaks
Ich verkaufe meine gesamte Software für VB 65 DM. Ihr Wert ist zur Zeit 4000 DM und steigt! M. Becher, Bombach 23, 5204 Lohmar 21

● VC 20 Spitzenprogramme ●
● Preiswerte Superprogramme, die ●
● auch wirklich laufen ●
● Liste von H. Santoro; Untere ●
● Mühlwiesen 11, 7896 Degernau ●

VC 20 Modulprogramme-GV, 3,8,16 K
Frogger + Centipede + Defender + Xeno + Scramble + Gridrunner + Donkey Kong + Traxx; geg. 20 DM (inkl. Kass. & Vers.) Horst Steiner, Ahornweg 5, 708 Aalen

★ Viele VC 20 Programme, Liste ★
★ gegen Ihre Liste oder Freium- ★
★ schlag bei Johannes Vogt ★
★ Straßburger Platz 6a ★
★ 2000 Hamburg 70, T. 040/616993 ★

VC 20 Grundv. Bundesligaprogr. erstellt Tab.-Progn.
5 DM an Knut Poburski, Irisweg 3 7407 Rottenburg 5

Verk. Spitzensoftware für alle Versionen des **VC 20**. Z. B. Original Modulspiele wie Gorf oder Schach, je nur 5 DM. Info für 1 DM bei C. Thiele, Schillerstr. 117, 1000 Berlin 49

★ VC 20 ★ VC 20 ★ VC 20 ★★
★ Super-Software, 300 Programme ★
★ GV 8K 8Ku (Rom-Module) 16K 16Ku ★
★ Superinfo geg. 0,80 Rückporto ★
★ B. Volk, Robertstr. 24, 5 Köln 91 ★

■■■■ VC 20 Software ■■■■
■ Große Auswahl, kleine Preise ■
Programme auf Kassette oder Disk
Auch Tausch! Gratisliste I. Möding
■ Engelswisch 32, 2400 Lübeck ■

VC 20, cbm 64 — tausche oder verkaufe SOFTWARE. Liste gegen 2 DM Rückporto. W. Geischläger, Troppauer Str. 38, 63 Gießen

★★★★ Ca. 400 VC 20 Prog. ★★★★
★ Kostenlose Liste anfordern bei ★
★ Wolfgang Pidun ★
★ Vorbruch 21, 4930 Detmold 17 ★
★★★★ Tel. 05231/88235 ★★★★

■■ Achtung VC 20 / C 64 Fans! ■■
■ Verkaufe 40/80 Zeichen-Karte ■
■ Suche außerdem C 64 Software ■
■ Robert Schuder, Ruhstorf 35½ ■
■ 8384 Simbach 1, Tel. 09954/682 ■

Biete Software an über 150 Programme. GV, Modul und 16 K. Liste gegen Rückporto oder Anruf bei Volker Dünzen, Moselstr. 10a, 5593 Pommern, Tel. 02672/1388, ab 14.00 Uhr

Drucke Ihre Listings. Copyright wird schriftlich garantiert.
Pro Seite (70 Zeilen) —,60.
Kassette oder Disk an H. Karlsson, Georgenstr. 49, 8000 München 40

VC 20 Modul Prog. u. a. Spiele-Forth, Graphic-Prog. Hilfe. 40 Zeichen, 30 verschiedene für DM 40 pro NN ohne Porto. P. Kryger, Bülten 11, Postf. 1249, 3078 Stolzenau

● VC 20 Superprogramme ● ab 1 DM ●
● Riesenangebot bis 16 K ● Info ●
● bei Udo Gnoss, Falkensteinstr. ●
● 293, 4200 Oberhausen. Gg. Frei- ●
● umschlag oder 1 DM Rückporto ●

Kaufe gute Software auf Kassette f. VC 20: 40/80 Zeichen ★ Deutscher Zeichensatz ★ Simon's Basic ★ Grafik ★ Dateien ★ Buchhaltung ★ Spiele. Angebote an: J. Deubler, 8861 Belzheim 8

VC 20: Wegen Systemwechsel verkaufe ich meine gesamte Software (vor allem Spiele). Programmliste anfordern bei: H. P. Wirth, Feldstr. 26, 6639 Rehlingen (Rückporto!)

VC 20 — Warum teure Bücher kaufen? Tips, Tricks, Spiele usw.
30 Prgr. gg. Einsendung v. 20 DM
A. Breitenbücher, Herm.-Allmers-Weg 20, 2870 Delmenhorst

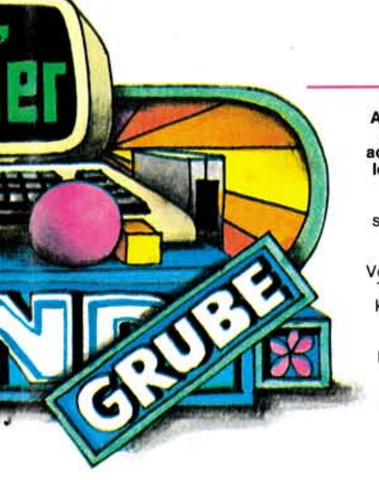
Suche VC 20 Freunde im Rhein-Sieg-Kreis um Erfahrungen auszutauschen. Josef Schneider, Feld 47, 5203 Much

VC 20: 30 Super Grundversionsprogramme für nur DM 19,—!!! Bei P. Hadorn, Steinerstr. 18, CH-3006 Bern. Tel. 0041/31/431162, ab 19 Uhr! Super-Software! VC 20!

DOPPELKOPF können Sie jetzt mit Ihrem VC 20 + 16 K oder VC 64 spielen. Der Computer ersetzt 3 Mitspieler. Kassette 20 DM, Disk 25 DM. Näheres unt. T. 0251/391991, ab 16 Uhr.

Systemwechsel! Verkaufe 100 Modulspiele (auf Kassette) für DM 200 auf Anfrage auch Pgm aus anderen Bereichen (GV, 3 K usw.). Info bei S. Müller, Sparkassenstr. 4, 4650 Gel.

13jährige Manuela — VC 20-FAN sucht preiswerten Monitor (evtl. 31 cm). Angeb. an: Manuela Petruck, Postfach 940153, 5000 Köln 91



Am besten verwenden Sie dazu die vorbereitete Auftragskarte am Anfang des Heftes. Bitte beachten Sie: Ihr Anzeigentext darf maximal 5 Zeilen mit je 32 Buchstaben betragen. Überweisen Sie den Anzeigenpreis von DM 5,— auf das Postscheckkonto Nr. 14199-803 beim Postscheckamt mit dem Vermerk »Markt & Technik, 64'er«, oder schicken Sie uns DM 5,— als Scheck, in Briefmarken oder in Bargeld. Der Verlag behält sich die Veröffentlichung längerer Texte vor. Kleinanzeigen, die entsprechend gekennzeichnet sind oder deren Text auf eine gewerbliche Tätigkeit schließen läßt, werden in der Rubrik »Gewerbliche Kleinanzeigen« zum Preis von DM 10,— je Zeile Text veröffentlicht.

!!! Ein neues Klangerlebnis !!!
Musikpr. VC 20/54 nur noch über Ihre Stereoanlage. Endlich Power!!
Interface steckerfertig 20 DM!!
Tel. 02871/46244

●VC 20●● Anwender-Software●●
● z. B. Rechnungsdatei ●
● Kostenl. Liste anfordern bei ●
● Ingo Molitor, Heckenstr. 29 ●
●5010 Bergheim 11●●●●●●●●●

VC 20 Spitzenprogramme für alle Ausbaustufen nur 3-6 DM oder Tausch. Liste gegen 80 Pf. Rückporto bei Werner Kleinhorst, Brücherhofstr. 34, 46 Dortmund 30

■ VC 20 und C 64 Besitzer! ■
Liste mit preisgünstigen Topprogrammen gratis. ★ Angebot: 50 VC 20 oder C 64 Programme für 59,— DM
■ A. Kohl, Pf 1138, D-6901 Hl. Kreuzst. ■

Verk. wegen Systemwechsel über 70 Prgr. f. VC 20 z. B. Frogger, Scramble, Galaxians, Grundv. u. Rom-Bereich. 10 Stck. 30 DM. Su. C 64 Prgr. G. Genech, Derner Str. 363, 4600 Dortmund 14

VC 20 Superpgme. zu Superpreisen z.B. 20 Module = 20,— DM u. a. Topangebote, schneller zuverl. Service. Info: Freiumschl. an G. Gerstmeier, Alter Postweg 84/I, 8900 Augsburg

---

VC 20. Verk. Spielsammlung u. a. Spielhallenhits w. Kong, PacMan, Crawler u. a. zum Teil mit Erweiterung, alle Spiele auf Kass. für 40 DM. F. Rupprich, Geysostr. 8, 3300 Braunschweig.

Wer liefert weißes Tabellierpapier für den VC 1515? — Morgenthaler, Federbachstr. 5, 7550 Rastatt 21

VC 20: 50 Spiele für Grundversion auf Kassette nur 20 DM. ★
Programme für GV auf Diskette: ★
0 Blocks free, nur 30 DM. ★
Auch Tausch! Tel. 06347/2747

+ + + VC 20 Softwaretausch + + +
Liste an: J. Bausa, Hirschweg 36, 5068 Odenthal, Tel. 02202/78499

VC 20 Programme hat A. Oldenburg, Lange Str. 47, 2190 Cuxhaven 12. Liste anfordern!

●VC 20●VC 20●VC 20●VC 20 ●
●Biete div. Hard- und Software ●
●Speichererw. Kassetteninterf. ●
●Info f. 1 DM von J. Kramke ●
●Otto-Suhr-Allee 135, 1 Bln 10 ●

VC 20. Tausche u. verkaufe Software. Prg. DM 1,—. INFO (Rückumschlag)! Bei Petruck, Postf. 940153, 5 Köln 91 Tel. nach 19.00 Uhr. Tel. 0221/864144

Gebe VC 20-Programmpakete supergünstig ab oder tausche. Info geg. Rückporto. M. Spurk, Danziger Str. 5, 8523 Baiersdorf

---

● Verk. VC 20 für DM 299,— ●
● oder im Tausch gegen —C64— ●
● Jürgen Knüttel, Ulmenallee 13 ●
●● 3062 Bückeburg, 5. Insp. ●●

VC 20 m. Drucker + 66 K Erw. + Progr.hilfe + 40 Z. + Assembler + Datasette + 2 Joyst. + 12" Monitor + Reset + Bücher + 50 Progr. — DM 1250 — auch einzeln. A. Höß, Pf. 111009, 8900 Augsburg

---

Verk. **VC 20**-Progr. für **1-4** DM!!
10-seitiges Superinfo m. Tips & Tricks o. Progr.-Kass. für 1,50 DM bzw. 3 DM bei ★ SKP (SW) St. Kaiser Cavalloweg 25, 7547 Wildbad 1 ★★

VC 20. Tausche Maschinenprogramme. Alle Speichergr.! Kass. mit Prg's an mich - Programme von mir zurück. Info gegen 90 Pf. ● Horst F. Müller ★Geisweiderstr. 143●5900 Siegen●

VC 20●Modulprogramme: 40 Stück für 40 DM/Kassette oder 50 DM/Diskette. Beste Spielprogramme/Utilities Info gegen 90 Pf. ● Horst F. Müller Geisweiderstr. 143, 5900 Siegen ●●

★★ CCL — Computer-Club-Laatzen
★★ bietet Ihnen 460 VC 20-Progr.
★★ auf 30 Disketten gegen cbm 64
★★ Postkarte an CCL, Eichendorff-
★★ str. 12, 3014 Laatzen 1

●●●●●●● VC 20 ●●●●●●●
Verkauf u. Tausch von VC 20-Prg.-GV, + 8, + 16. Liste gegen Rückp. an H. Dietrich, In der Schanz 14A, 6506 Nackenheim, Tel. 06135/4044

VC 20-Software aus England.
Liste + Gratisspiel nach eigener Wahl bei U. Hein, Christian-Kruse-Str. 19, 2300 Kiel 1, Ps. Preis ca. bis 4 DM

Suche schnelles Disketten-Kopierprogramm für VC 20 + VC 1541 Floppy + 16 KB. Das Prg. sollte mit einer Floppy laufen. Udo Grün, Kirchenfelder Weg 40, 56 Wuppertal 11, Tel. 02058/80008

Programme für die GV des VC 20? Superstarke PRG 2 DM!!
Liste gegen 80 Pf. bei Harald Arndt, Ringstr. 19, 8401 Alteglofsheim. Tausche auch Programme

---

VC 20-Besitzer aus Österreich sucht alle möglichen Programme und Spiele, aber auch Bücher.
Info bitte an: ● David Gibus, Stechergasse 17, A-4025 Linz a.d. Donau ★

11jähriger VC 20-Fan sucht sehr preiswerten Drucker und Floppy (habe nur wenig Taschengeld). Torsten Reiners, 2190 Cuxhaven, Elsa-Brandström-Str. 2

Blitzversand VC 20 / VC 64 PGME
● Sofort freies Info anfordern ●
Umfangreiche Programm-Sammlungen von H. Wolf, Tangstedter Str. 5, 2080 Pinneberg, Tel. (04101) 27293

VC 20/cbm 64: Liste mit 1500/200 Prgr. gegen DM 1,30 in Briefm. Tausch oder billig, Kass./Disk. M. Herrijgers, Körtingsd. Weg 50, 3000 Hannover 91

★ FUNDGRUBE ★

■ VC 20 Programmtausch z.B. 70 ■ Module + Adventures + Utilities ■ über 350 P. von GV bis 64 K ■ Liste 1 DM in Bfm. Chr. Eichhorn Eichendorffstr. 12, 3014 Laatzen 1

Suche 27 K o. 16 K RAM m. Steckpl.Erw. 2fach; Skatprogr. 8 K o. 8 KU; Bauanleitungen m. Printplan für VC 20 Zub. Angebote an: Meyerhoff, Postfach 1242, 6093 Flörsheim

Günstige Gelegenheit: Programme für VC 20 auf Kassette o. Disk, kostenlose Info gegen frank. Antwortkuvert. H. Stadelmann, Am Steigacker 1, 8501 Schwaig 2

Verk. VC 20 + Kass.-Interface + viele Programme auf Kass., Listings + Literatur, neu 900,-. Jetzt VHB 495,- wg. Systemwechsel. J. Grzescik, 2418 Ratzeburg, Posener Str. 15

● VC 20 ● VC 20 ● VC 20 ●
● Super Hot Games Info (80 Pf. ●
● in Briefm. beilegen) anfordern! ●
● Abs.: Matthias Stelling, Dorf- ●
● str. 148, 2126 Adendorf ●

Original **VC 20-Programmkassetten**
Neu! original verpackt nur 150 DM!!
Kanone/Luftkampf/Biorhythmus
Datasette neu nur 100 DM! J. Atrott, Weidenauerstr. 142, 59 Siegen

Tausche VC 20-Programme. Anrufen oder Liste an:
J. Gutzke, Dessauer Str. 54, 4050 Mönchengladbach 1
★★★ Tel. 02161/181306 ★★★

VC 20-Programmkassette, 10 Spiele für Grundversion 29 DM. Bestellung an: Postscheckkonto 249793-800 PSchA Mchn. Manfred Riedlberger, Vogelmauer 31, 8900 Augsburg

■ **VC 20-Spitzensoftware:** ■
■ Progr. für alle Erweiterungen!! ■
■ Modulprogr. nur 2,50 DM! Info ■
■ geg. Rückporto bei: M. Koch, ■
■ Jahnstr. 160, 8300 Altdorf ■

● VC 20 Österreich VC 20 ●
● 60 Modulprogr. auf Kassette!! ●
● zu günstigem Preis sowie 200 ●
weitere Progr. zu verk. K. UNGER Ob. Hauptstr. 35, A-7162 TADTEN

Programm-Listings für Ihren VC 20 von Kass. o. Disk drucke ich für eine Gebühr von 20 Pf pro DIN-A4-Seite (+ Rückporto). Udo Grün, Kirchenfelder Weg 40, 5600 Wuppertal 11

VC 20:Programmtausch und Kauf Suche auf Kassette +32 K gutes Fußballspiel, Flugsimulator, Formel I (Raider), Skat (VC 20 = 2.+3. Spieler). Tel. 0451/23237, auch am Tage

★ VC 20-Software - große Auswahl ★
★ Tausch und Verkauf. Info gratis ★
★ von Michael Rosenkranz ★
★ Wurster Str. 13, 2800 Bremen 66 ★
★ Tel. 0421/511228 (ab 18 Uhr) ★

●●●● Commodore VC 20 ●●●●
● 280 Superprogramme superbillig ●
● Verk. 8/16 K-Progr. für 55 Pf. ●
● Verk. auch 8 K-Ram + Progr. für ●
● die Grundversion. Tel. 06772/6498 ●

Wer hat Zaxxon für den Commodore VC 20! Anbieter bitte melden bei Jan Krüger, Auf dem Kamm 6a, 5060 Bergisch Gladbach 1, Tel. 02204/65376

Verkaufe Softw. für VC 20! Z. B. Defender, Seawolf, Crazy Kong etc. 100 Pr. stehen zur Auswahl! Info gegen 90 Pf. Jörg Gericke, Wiedhof 15, 2220 St. Michaelisdonn

●●● VC 20 Superangebot ●●●
● 100 Pgm's für die Grdvers. ●
● mit Anltg. auf 2 CC DM 49,— ●
● NN od. V-Scheck. M. Rudolph ●
● Elisabethstr. 96, 23 Kiel 14 ●

Superspiele Tausch + Abgabe VC 20, GV + 8 K + 16 K
Verkaufe VC 20 + 8 K + 16 K Literatur, Schachmodul, VHB 580,— DM. H. Römer, Enteng. 16, 7129 Pfaffenhofen

COMMODORE 64

Achtung VC 64 Besitzer!!! Biete Software! Die besten Spiele! Billigst! Nur Disk! Umfangreiche Liste gegen 1,30 DM. B-Soft, Homburger Str. 35, 6365 Rosbach

Commodore 64 Superprogramme!!! Spiele wie Benji 3-D PacMan billig. Liste gegen 1 DM in Briefmarken. J.-P. Groene, Allenberg 12, 5788 Winterberg 2, Tel. 02983/1464

Suche und tausche Amateurfunksoftware für Commodore 64
Wilfried Bestmann DL8 HAV
Kieler Str. 6, 2081 Bilsen
Tel. 04121/232393 oder 04106/2313

●▲● cbm 64 Supersoftware ●▲●
▲ Spitzenprogramme! Tausch oder ▲
● Verkauf gegen Mini-Unkos.-Beitr. ●
▲ Liste (80 Pf) ■ Jürgen Wehner ■▲
● Eichenweg 5 ■ 4722 Ennigerloh ●

SUCHE SOFTWARE — Spiele + Nutzprogramme.
Bitte Ihr Angebot an:
Ulrich Seck, Schulkirchweg 16
4300 Essen 1

★★★★★★★ C 64 ★★★★★★★
★ Tausche Top-Programme für
★ C 64. Liste, nur gegen Rück-
★ porto, von: V. Marhs, Herder-
★ Straße 6, 7100 Heilbronn

VC 64: 75 Masch.-Games auf Kassette für 100 DM. Kein Schund, alles Original. Tel. 030/6121312

★★★ C 64 Softwaretausch ★★★
Tausche C 64-Programme auf Disk oder Kassette. Liste an:
B. Hauptt, Heilsterweg 6,
2382 Kropp, Tel. 04624/8728

Suche Software aller Art mit deutscher Anleitung. C 64 User zwecks Austausch im Raum NR-AK gesucht. Helmut Rose, Wilhelmstr. 7, 5419 Raubach, Tel. 02684/4881

■ ★ Commodore 64 ★ Erfahrungs- und
■ Programmtausch, Anruf ab 17.00
■ (Kreis Herford) Liste an: HARALD
■■■ BLOEBAUM, Strotweg 22, 4983
■ Kirchlengern 4, Tel. 05744/2371

★★★ Der Commodore 64 Club ★★★
Superprogr. von Simon's Basic bis Frogger, Text od. Datenverw. Liste bei P. Schwarze, Kasslerstr. 8, 3549 Breuna, Tel. 05693/7007, 9-16 Uhr

★★ Verkaufe Commodore 64 ★★
neuw., incl. vielen Programmen;
M. Gaksch, Trettachstr. 68
8900 Augsburg (tel. Rückruf erfolgt sofort)

X cbm 64 Software - auch Tausch X
X 80 Zeichen/Zeile auf dem TV X
X 136-Farben-Turbotape, alles 20 X
X DM. Günter Gensler, 8732 Mün- X
X nerstadt, Schwimmbadweg 5 X

C 64 Software auf Kass. o. Disk

Dt. Zeichensatz f. C 64	20 DM
VIC 1515/25 als Schreibm.	30 DM
Textverarbeitung	50 DM

Schumacher, Ahornweg 18, 53 Bonn 2

VC 64 Wertpapier-, Zinsen u. Mathe-Programme DM 12,— auf Disk DM 16,—. Vorauskasse a. Postscheck München 113028-800 oder Scheck an Peter Fabri, Langwied 69, 8000 München 60

★ FUNDGRUBE ★

★★ SUPERSOFTWARE-Pitstop ★★ Nur 1A-Qualität, 20seitiger Katalog gegen 1,20 DM bei M. Brox, Ringstr. 26, 3579 Frielendorf 5, Tel. ★ 05684/7347 ★★★ SUPER ★★★★

●C 64●z. B. **20 TOPGAMES**●C 64● (Grandmaster, Pool, Pac, Jaw, Frog, Skramble, Camels ...) öS 500, DM 70 ● Stefan Schöpf, Forchach 8 ● ●● A-6166 Fulpmes/Österreich ●●

★★★ Recorderinterface f. VC 20/64 cbm 234/8000 Fertiggerät im Geh. m. allen Steckern 39,— DM, Bausatz 27,— DM. Klaus Oelker, Hoye 1, 2225 Schafstedt, Tel. 04805/380

Biete ca. 150 Spiel- und 50 Anwenderprogramme für den VC 64 an (z. B. Simon's Basic, Schach, Frogger usw.). Gratisliste anfordern bei: Postlagernd Nr. 012108 B, 1000 Berlin 12

●●●● **COMMODORE 64** ●●●● ● Tausch od. gegen Unkosten Info ● ● bei Urs Bertschy, Eichwaldstr. 9, ● ● CH-6300 Zug, Tel. 0041/42319509 ● ●●●●●●●●●●●●●●●●

●●●● COMMODORE 64 ●●●● Top-Programme, Tausch oder geringste Unkostenbeteiligung. Infos bei Michael Leuschner, Eenstock 12, 2000 Hamburg 71

●★★★SUPERSOFTWARE★★★● ● Eine ganze Diskette randvoll ● ● mit Superprogrammen. Liste ● ● gegen Rückp. W. Armbruster ● ● Dorfstr. 7, 7601 Ohlsbach ●

★★★ **cbm 64 SOFTWARE** ★★★ Nur Spitzenprogramme! Tausch bevorzugt, Kauf-Verkauf ca. 5 DM je Programm. M. Breiing, Welscher Heide 2, 5060 Berg. Gladb. 1

★■ C 64 — SUPERANGEBOT ■★ ★■ 1. Diskette Programme ■★ ★■ nach Ihrem Wunsch ■★ ★■ 40,— DM (Scheck) ■★ ★■ ABSPRACHE 02173/78343 ■★

■■■■ Commodore 64 ■■■■ Super-Programme — Super-Preise <15 DM! Auch Tausch! Info gegen ■ 80 Pf. bei S. Krücker, Stettiner ■ ■■ Weg 20, 5309 Meckenheim ■■

cbm 64★SUPERSOFTWARE★cbm 64 Programme jeder Art, z.B. Frogger Schach ★ Motor Mania ... kostenlose Liste: Klaus Kappert, Postfach 1208, 8980 Oberstdorf ★ Auch Tausch !!!

C-64 Riesige Top-Prg.-Sammlung (Hobbit, Simons, Jumpman, Benj., Pilot, Logo, Forth etc.) NP 4000.- für 300, auf Disk bei Michael Börner, 6719 Weisenheim, Im Winkel 8, Tel. 06353/8625

●●●● COMMODORE 64 ●●●● ● Über 300 Prg. aller Art. Tausch ● ● oder Unkostenbeitrag. Liste ge- ● ● gen Fr. 1,60 in Briefm. bei D. Winter- ● terberg, Altbachstr., CH-8305 Dietlikon

Tausche 3 sehr gut fliegende Modellflugzeuge für Elektroantrieb gegen Floppydisk-Laufwerk VC 1541 oder Drucker VC 1525: C. Wiese, Kirchstr. 13, 2730 Heeslingen

cbm 64 Verkaufe + Tausche Prg. zu fairen Preisen u. 10 DM. Bei Tausch bitte Tauschliste od. Anfragen an R. Hiddemann, 4600 Dtmd 41, Märker Grund 7, Tel. 0231/486818 schicken.

■■■■■■■■■■■■■■■■■■ ★ **Spitzenprogramme für den C 64** ★ ★ Biete tolle Programme gegen ★ ★ gebrauchte Wikingautos. Roland ★ ★ Koch, Schwanenweg 7, 2308 Preetz ★ Tel. 04342/4325, ab 19 Uhr. ★ ■■■■■■■■■■■■■■■■■■

cbm 64 + VC 20 Spitzen-Software! Biete tolle Programme gegen gebrauchte WIKING-Autos. Roland Koch, Schwanenweg 7, 2308 Preetz, Tel. 04342/4325

Neueste VC-64-Software (günstig abzugeben, Disk. o. Kass.) GCSB Fröbelweg 7a, 2000 Norderstedt, Tel. 040/5224292 (17-18 Uhr) (mögl. schriftl.). Falls Liste 1 DM beilegen

●●●●●Commodore 64●●●●● ● Tausch + Kauf Liste an: ● ● V. Terber, Röttler Ring 13 ● ● 7889 Grenzach-Wyhlen 1 ● ●●●●●●●●●●●●●●●●●

★★★★★★★★★★★★★★★★★ ★ Suche gute 64er-Software für ★ ★ soz.-wiss. Statistik, (ähnlich ★ ★ SPSS) sowie Programm für wiss. ★ ★ Bibliographien (Literaturdatei- ★ ★ Verwaltung). Angebote an: ★ ★ Dr. Rudolf Egg, Budapester Str. ★ ★ 16, 8520 Erlangen ★

C 64. Sehr gute Software auf Disk./Kassette. Liste gegen Freiumschlag! Szczepanski K., Semmelweisstr. 6, 5090 Leverkusen 1

Superprogramm für C 64. Schützt Dateien vor »scratch«, formatiert Disketten, holt gelöschte Dateien zurück, löscht Diskette. **Diskette DM 50,—**. Frank Barte, Nordstr. 37, 3300 Braunschweig

cbm 64. 90 Top-Programme gegen DM 50,— Unkostenbeteiligung a. Kass./Diskette. Best. per VS o. bar an M. Günsche, Odenwaldstr. 13, 6980 Wertheim 2 (Info DM 2,- in Bfm.)

■ VC64-Spiele + Gesch.-Programme ■■ Das Neueste aus den USA ■■ ■■ über 400 Progr. ab DM 5,— ■■ ■■ Kostenlose Info von M. Marx ■■ ■ Schlesische Str. 8, 4939 Steinheim

C64-Anwenderclub München, Justus Erb, Theresienhöhe 6b, 8000 München 2, Tel. 089/5023659

VUP-Software für cbm 64 Datei-Verwaltungs-Systeme, Games, Fixings. Liste 2,— DM. Heinemann, Talstr. 18, 6057 Dietzenbach 1, Tel. 06074/32949

● **cbm 64:** Tauschen und Verkaufen ● ● Programme. Liste gegen 80 Pf. ● ● Rp. (1 DM) in Brfm. bei M + R. Lutz, ● ● Tannweidstr. 3, CH-8133 Esslingen● ●● **Günstige Preise und Rabatt!!** ●●

Daten-Plot-Progr. zur graphischen Darstellung versch. Meßwerte für Comm. 64 schriftl. Informationen u. Preis: Claus Röhrborn, Jägerschneise 28, 6307 Linden, Tel. 06403/61266

Tausche Software für VC 64 (c. 350 Programme) oder Zeitaufwandentschädigung. Schickt Eure Listen an Thomas Preul, Steilshooper Str. 183, 2000 HH 60, Tel. 040/6919454

●● $$ **Commodore 64** $$ ●● ● Tausche und verkaufe 64er ● ● Prg. Habe die allerneuesten. ● ● Liste an: ★Karsten Bettels★ ● ● Rahewinkel 46, 2 Hamburg 74 ●

★ Suche: VC 64 für ca. 300 DM ★ TAUSCHE VC 20 SOFTWARE, Liste an: A. Fieber, Eichenweg 8 8098 Pfaffing Tel. 08076/575

VC 64, verkaufe meine gesamte Software für 150 DM, Wert 10000 DM, 15 Disketten oder auf Kassette, Telefon 0208/840811 ab 19.00 Uhr

Suche für Commodore 64 Spiele u. Programm für Abnahme-Diät. Angebote an: L. Wacker, 5350 Euskirchen, Memelstr. 3

●●●●●● C 64 ●●●●●●
●●● Wasserbauprogramme ●●●
Dipl.-Ing. Olaf Müller -WG 9- Universitätsstr. 19, 4300 Essen 1, Tel. 0201/231685

? Wer verkauft mir C64-Zubehör? Z.B.: Disketten-Laufwerk, Drucker ... sowie: Pascal + div. Software-Angebote; bitte an J. Pöhler, 6053 Obertshausen, Rosenstr. 17, Tel. 06104/72010

Suche Software aller Art für den C 64 sowie Kontakt zu C-64-Usern auch Tausch und Verkauf. Hanns Günther, Schulteß Damm 32, 2 Hamburg 65, Tel. 040/5366818

Tausche Software für C 64. Wer hat Interesse am Anwender-Club im Raum Bremerhaven? Günter Struck, Mercatorstr. 21, 2850 Bremerhaven, Tel. 0471/85369

●●● cbm-64-Software ●●●
Alle DM 6,—; Liste DM 1,50; Klaus Schiller, Neuenburgerstr. 21, 1000 Berlin 61

TABELLIERPAPIER
alle Größen sofort lieferbar: TAB-Preisliste MF anfordern.
RAUSCH DRUCK GmbH, D-8900 Augsburg, Pf. 102304, Tel. 0821/77091, Tx 53785

★ ★ cbm 64 und keinen Drucker? ★ ★ -Nicht schlimm- ★ Drucke Ihr Prg. für 30 Pfg./Blatt + Porto ★ Kass. oder Disk an Manfred Auer, Alpenrosenweg 17, 8972 Sonthofen

★ cbm 64 X Software ★
★ Tausche Prg. aller Art ★
Andreas Mellen. Spiekeroogerstr. 28 2948 Schortens 1, Tel. 04461/80740

Software für Schachspieler:
Partienarchiv, Eröffnungstraining usw. Info für 80 Pf. Rückporto. V. Stürcken, Zedeliusstr. 20, 2940 Wilhelmshaven

VC 64 ★ Verkauf ★ VC 64 ★ Tausch ★ Jedes Programm 6 DM ★ ★ Disketten oder Kassetten ★ ★ Liste für 1.30 DM ★ ★ Thomas Doll ★ ★ Severinswall 16-20 ★ ★ 5000 Köln 1 ★ ★

Musik mit dem C 64? Maschinenprg. ermögl. das dreistimmige Spiel beliebiger Musikstücke! Prgr. mit Demostück auf Kass. nur öS 300. A. Schneider, A-9920 Sillian 1 h

● Super cbm 64 Software ●
● Tausch + Verkauf ●
● Liste gegen Freiumschlag ●
● S. Schramm, Rembrandtstr. 5 ●
● 6074 Rödermark ●

Suche: Erklärung für Busy-Calc., kommerz. Programme, PET-Speed-Compiler und Spiele (nur MSP.)
Michael Meister, Am Hellweg 8B, 3490 Bad Driburg

Suche Software für C 64 Statistik (Refa-Arbeitsstudium), Elektrotechnik sowie Materialwirtschaft.
Richard Gottschalk jun.
Grenzstr. 3, 4154 Toenisvorst I

Suche gut erhaltenen, gebrauchten VC 64 bis 400 DM.
J. Moczko, Martin-Luther-Str. 6, 2000 Hamburg 11, Tel.: 040/371399, ab 18.00 Uhr.

VC64 ★ Wegen Systemwechsel ★ VC64 verkaufe ich meine komplette Softwaresammlung auf Disk + Kass. Info gegen DM 1,-. J. Erdmann, Rochusstr. 187, 5300 Bonn 1

Raum Nürnberg — Suche und tausche Spielprogramme für Commodore 64,
Tel. 0911/460484, ab 14 Uhr

C64-Superprg. Tausch u. Verkauf. (Verk. jedes Prg. gegen geringen Unk.-Beitrag). Wenn mögl. Tauschliste beilg. Info bei: SOFTEX 64, Postfach 3440, 6300 Zug 3/Schweiz

Suche günstige 64er-Software-Programme, Spiele und Hilfsprogramme — auch günstiger Drucker gesucht, gebraucht oder neu — Herbert Mayer, Neideggweg 55, 7906 Blaustein

★ C64. Suche Tauschpartner für ★
★ C64-Software! Egal ob Kasset. ★
★ oder Diskette. Martin Amelang, ★
★ Schleifmühlenweg 28, 6082 ★
★ Mörfelden, Tel. 06105/23481 ★

★ VC 20-Superprogramme ★ (Exbasic, Choplifter etc.) sehr billig. Info gegen Porto (auch Tausch). M. Amelang, Schleifmühelnweg 28, 6082 Mörfelden

cbm 64: Jede Art von Software u.a. Spiele, User-Prg. usw. demnächst: ZAXXON, Gratis-Information bei A. Schladitz, Pillnitzer Weg 33A, 1000 Berlin 20

C 64 ★ 300 Programme aus allen ★
Bereichen ★ ab DM 5,— ★ Katalog ★
DM 3,— in Briefmarken ★ ★ ★
Treichel, ★ Mittelbruchzeile ★ ★
105 ★ D-1000 Berlin 51 ★ ★ ★ ★

cbm 64 SOFTWARE verkaufe gesamte Software = 30 Disk (über 400 Prg). Disks auch einzeln abzu.; Gratis-Info: G. Schuller/Untere Klaus 95/A-8970 Schladming

★ cbm 64 ★ 1 Diskette voll mit MC-Spielen (Frogger, Pacman usw.) 40,— DM in Scheinen. Auf Kassette 30,— DM; Frank Scholz, Beim Brückle 6, 7401 Nehren

Suche Progr. für den cbm 64, bes. Spiele, mathem. chem. und phys. Progr., suche auch gebr. Farbmonitor und Floppy VC1541. M. Offermann, Junesrotstr.7, 5020 Frechen 4

Angebot Donkey-Kong Camel Skramble Best auf Kassette zusammen 20 DM, auch Tausch anderer Programme auf Anfrage; M. Wagner, Ruetten-Schneider Str. 63, 4300 Essen 1

■ Kaufe—Tausche—Verkaufe ■
cbm 64-Programme auf Diskette Angebote/Anfragen (Rückporto)
■ G. Riha, In den Unterwiesen 20 ■
■ 6239 Kriftel, Im Vordertaunus ■

■■■■■■■■■■■■■■■■■■
■ ★ **64 FORTH** ★ f. C 64. Eine neue ■
■ schnelle Sprache, unterst. Graph. ■
■ u. Sound, VC64. Info b. Dral-D. Luda ■
■ 8000 Mchn 83, Staudingerstr. 65 ■

● ★ ★ COMMODORE VC 64 ★ ★ ●
● Superprogramme — Info gratis ●
● 25 Top-Spiele 20,— DM (Schein) ●
● auf Diskette oder Kassette ●
● Rutz, Postf. 7036, 5810 Witten 7 ●

●● C 64 Systemwechsel Superspiele: 25 Disk. à 40,—. Flipper, Jumpman, Loderunner und weitere Supersoftware zu Discountpreisen. F. Burghardt, Saarstr. 29, 6360 Friedberg

★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★
Suche für meinen cbm 64 ein gutes Skatpgm. mit Null + Ramschspiele O. Griess, Weidenbaumsweg 93A 2050 Hamburg 80

+ + + Suche Jugendliche + + + + die mit mir Programme und Erfahrungen auf dem Commodore 64 austauschen + + + Dietmar Gördel + + Konsul-Lorentzen Str. 9 + 2340 Kappeln

FUNDGRUBE 

DIRSORT bringt Ordnung in Ihre Programmsammlung — alphabetisch geordnete Liste der Directoryinhalte aller eingelesenen Disketten in wenigen Sekunden mit Ausgabe auf Drucker oder Bildschirm — auf Disk DM 15,— C 64, PF 819, CH-8021 Zürich

Suche Pgme. für C 64! Auch gegen Unkostenbeteiligung! Außerdem suche ich Mitglieder zur Gründ. eines C 64-Clubs Ansbach. R. Geyer, Sudetendeutsche Str. 13a, 8800 Ansbach

Commodore 64 Software — Tausch oder Unkostenbeteiligung. Billiger und schneller geht's nicht mehr. Info von Bernhard Hoffmann, Am Harras 6, 8000 München 70

★★ cbm 64 Software ★★ zu Niedrigstpreisen abzugeben (z.B. Simon's Basic DM 15, Jumpman DM 12, usw.), Pierre Weimerskirch, 74 Kohlenberg, L-1870 Luxemburg ★ Info gegen DM 1 ★

Suche, tausche Programme für C 64 auf Kassette oder Listing. Wo gibt es den billigsten Drucker? Peter Moschko, Obere Burachstr. 25, 7980 Ravensburg

Tausche Programme für cbm 64 oder gegen Unkostenbeteiligung. Tel. 06421/44185, ab 17 Uhr

★★ Roulette ★★ Das besondere Programm, tolle Grafik + Sound Diskette oder Kassette 40,— DM Info gegen Rückporto. E. Meyer, Pfaffenmauer 31, 6000 Frankfurt 60

★★★★★ HOTWARE ★★★★★ 64er Supersoftware! Z. B. Donkey Kong, Soccer...! Schon ab 5,— DM! Info gegen Freiumschlag bei: Lars Grüne, 48 Bielefeld 15, Bruchweg 18

★★ cbm64 Haushaltsprogramm ★★ 11 Konten, ca. 1000 Buchungen pM. auch ohne Floppy komfortabel DM 20,—. Info gg. DM 0,80. F. Hornung Seewiesen 1, 7410 Reutlingen 28

13011984130119841985/1985198 3198219811980 = ? Kaufe Druckerprogramm mit beliebig vielen Nachkommastellen. A. Wordell, Helene-Mayer-Ring 14/1808, 8000 München 40, Disk

VC 64 Superangebot — 30 Pgm's auf CC DM 45,— NN od. V-Scheck. M. Rudolph, Elisabethstr. 96, 23 Kiel 14

★★★ C 64 EPROM PLATINE ★★★ doppels. u. verz. Epoxydharz, 8K. Eprom 2532 o. 2564, \$8000-\$9FFF. 30 DM Tel. 02331/51807

★★ VC 20/C 64 RESET-TASTER ★★ ★ Steckergehäusxe, kein Einbau! ★ DM 10,—, incl. Porto u. Verpack. ★ Tel. 02331/51807, ab 19 Uhr ★

Sorgfältig getestete Beispiel-Programme zum Commodore 64. Beschreibung, Listing, Kassette bzw. Diskette. 8 Programme: 15 DM. M. Heyn, GHB021003, 7743 Furtwangen

Gesucht: Renumber-Routine mit Gosub, Goto-Änderung. Wer hat Erfahrung mit Practicalc und Calc Result und ähnl. Programmen. M. Heyn, GHB021003, 7743 Furtwangen

Optimiertes Basic auf dem VC 64, ein Kurs an der VHS Furtwangen. Auch Sie können die begleitenden Unterlagen erhalten. Info von M. Heyn, GHB021003, 7743 Furtwangen

cbm 64 Programme 06421/44185, ab 17 Uhr

Super-Angebot, riesige Programmsammlung, NP über 4000,—, fast alles, was es für den C 64 gibt DM 300,—. H.-J. Scheidt, Karl-Gayer-Str. 2, 6719 Weisenheim, Tel. 06353/8673

C 64 — Habe 25 Disk. voll m. Prgm. Tausch o. Verkauf, Liste an: I. Molitor, Heckenstr. 29, 5010 Bergheim 11

VC 64: Einmaliges Angebot. Über 300 Programme aus allen Bereichen für nur 100, — DM. Wer nicht zugreift, ist selber schuld!! Tel. (030) 6121312

★ Commodore 64 ★ VC 20 ★ Die wirklich interessanten Pgm. bietet Ihnen: Tel. 0221/432457

TEXT 80-Textverarbeitung für 80 Zeichen (Disk DM 15,—) VIDEO 64-Videothekverwaltung (Disk 10,—) oder Info (1,—) C 64 PF 819, CH-8021 Zürich

Joystick f. VC 20, CBM-64, Atari DM 29,50; Quickshot 49,50 DM, Prg. für cbm-64 ab 2.— DM (auch Tausch). Liste gegen Rückporte. G. Mantwill, Dorfstr. 47, 2251 Wobbenbüll, Tel. 04846/1570

cbm 64 Super-Software VC 20 — spottbillig, Info gegen Freiumschlag. Tausch erwünscht. Horst Buhl, Neue Gasse 42, cbm 64 872 Schweinfurt VC 20

Kassetten C 20 = 2 x 10 Min. mit Box u. neutralen Aufklebern im 10er Pack. DM 23,—. Nur Vorkasse Pschk. Essen 185355-431 oder V-Scheck. F. P. Doerr, 6251 Burg — Balduinstein

★ Top-Software VC 64. Neueste u. aktuellste Programme zu guten Preisen; mit Mengenstaffelung ★ K. Schünemann, 4600 Dortmund 41, Marker Grund 7, 0231/486818 ★

Commodore 64 — Programmaustausch auf Disk. Liste gegen Freiumschlag. Artur Engels Griesbergerstr. 64, Tel. 02174/62272 5093 Burscheid (Raum Köln)

Tauschpartner/in für C 64 Programme (nur Diskettentausch) gesucht. ★★★ Anfragen unter Tel. Nr. (07123) 7513, App. 18 (tagsüber), auf Wunsch erfolgt Rückruf!

VC 64 Programmtausch — Suche zuverlässigen Tauschpartner für längeren Zeitraum — 350 Pr. vorh. — Liste an: Thomas Doll, Severinswall 16-20, 5000 Köln 1

C 64 — Erstelle Prg. nach Wunsch. Anfragen bitte an: I. Molitor, Heckenstr. 29, 5010 Bergheim 11

Supersoftware tauscht Preul, Steilshooper Str. 183, 2 Hamburg 60. 040/6919454. Evtl. auch Zeitaufwandsentschädigung! Schickt Eure Listen und Wünsche an mich!!!

VC 64 Programmverkauf — Jedes Programm unter 6 DM / 350 Progr. vorhanden / evtl. Kassette — Liste geg. 1,30 DM: Thomas Doll, Severinswall 16-20, 5000 Köln 1

Habe C64—Bin Neuling im Raum Basel, suche Gleichgesinnten oder Club. Wer verkauft Drucker für cbm 64. Fritz Wanner, Fraumattstr. 39, CH-4410 Liestal, Tel. 061/945429

★ FUNDGRUBE ★

C 64 SPITZENSOFTWARE AUF DISK abzugeben; PETSPEED-Compiler 55-; Austro-Comp. 50-; 9 Spiele Donky Kong 3D-Froger u. v. m. insgesamt 59.- F. Brall, Wollweber 9, 643 Hersfeld

Täglich neue Spitzenprogramm - stop- Liste anfordern -stop- 1 DM in Briefmarken (für Unkosten) -stop- H. Kosfelder, Adelgundenweg 65, 4300 Essen 1

■!■!■!■!■!■!■!■!■!■!■!
Schüler sucht billige VC 1541! Tausche auch C 64 PGM.: Liste an: Volker Mücke, Im Hag 32, 5180 Eschweiler, Tel. 02403/4230 – – – –

C 64 Anwenderclub München, Justus Erb, Theresienhöhe 6b, 8000 München 2, Tel. 089/5023659

■ VC 64 Software für helle Köpfe ■
■ Superbillig oder auch Tausch ■
■ Top-Auswahl wie z. B. Jump- ■
■ man, Simon's-Basic, Fußball, ■
■ Soccer usw. Tel. 0471/37151 ■

cbm 64 Supersoftware-Superpreis - Verkaufe gegen Unkostenbeitrag Liste 32 S gegen 1,60 Briefm. bei G. Kraschewski, Longbentonstr. 60, Super - 4353 Oer-Erkenschwick - Super

★★★★★★★★★★★★★★★★★
Commodore 64 Software **bis 5,— DM** Schreiben Sie uns: 64er User's, Tucholskystr. 112, 8500 Nürnberg
★ **1,90** in Briefmarken beilegen ★

●●●●●●●●●●●●●●●●●
● Tausche C 64 Software! Liste ●
● bitte an: C. Elsholz, Frieling- ●
● hausener Str. 42, 5063 Overath ●
●●●●●●●●●●●●●●●●●

● Verkaufe Vokabel- u. Über- ●
● setzungs-Programm (Kassette) ●
● für nur 15 DM (Scheck/Schein) ●
● Bezug über: J. Martin, ●
● Achalmstr. 3, 7310 Plochingen ●

COMMODORE 64
Suche Programme für VC 64. Liste o. Angebote an: Ralf Siebert, Falkenweg 1, 3253 Hess-Oldendorf 2

ACHTUNG! cbm 64-SOFTWARE gegen Unkostenbeitrag oder Tausch. Liste gegen Rückporto (80 Pf.) von: Sven Emmerich, Agnes-Miegelstr. 34, 4030 Ratingen 8

Das ist der Hammer! Die besten Prg's aus aller Welt für cbm 64 und VC 20, superbillig. Powersoft, Postlagerkarte Nr. 098858A, Postlagernd 23 Kiel 1; Tausche auch!!!

Suche günstige Tastatur für cbm 64 (evtl. defekten VC 20 oder 64). Hans Schmidt, Tel. 06381/3216, ab 18 Uhr 06381/8202

C 64: Die Alternative zur Floppy! Fasttape macht Datasette 10 mal schneller. Auch Autostart. Nur 20 DM. Kein Error. Info 80 Pf. R. Heinrich, Bertastr. 20, 4650 Gelsenk.

Suche cbm 64 + Floppy (Monitor) günstig (auch Software).
B. Stopp, Wittinger Str. 101
31 Celle. Angebote bitte schriftlich mit Tel. Nr.

● VC64 Spitzenprogramme werden ●
● gesucht + getauscht. Sofortige ●
● Rückantwort bei Einsendung von ●
● Tauschlisten an Robert Dahlen, ●
● Exiner Str. 3a, 1000 Berlin 47 ●

★★★★★ SOFTWARE ★★★★★
★ Verk. meine gesamten VC64 Prg. ★
★ für nur 120 DM. Stefan Hansmann ★
★ Ritterbüschel 6, 673 Speyerdorf ★
★★★★★ 06327/4423 ★★★★★

Suche gebrauchten, neuwertigen VC 64 mit Datasette, Lightpen, TV-Anschl. + Software. Preisangebot ex Raum München an Tel. 089/8418375

★ VC 64 ★ VC 20 ★ Die neuesten Spiele und Programme! Disketten- und Kassettenversand. Info gratis bei Rainer Müller, Tel. 0511/612130, Reimerweg 8, 3004 Isernhagen

VC 64 Spitzenprogramme günstig! Spiele, Datenverarbeitung, Keyboard. Über 100 Programme in Maschinensprache. Volker Maack, Breitergang 7, 2 Hamburg 36, Tel. 040/343816

Superspielesammlung DM 49,50
Hungry-Horace DM 24,50
Lücker H. Eisenbach 37, 627 Idstein

C 64-User wünscht Erfahrungs- u. Software-Austausch! Angebote über Hard- u. Software (keine Spiele) an: Harry W. Hertz, Pf. 2664, 675 Kaiserlautern

★★★ cbm 64 + Floppy 1541 ★★★
Diskettensortierprg. Liest alle Prog. von Disk ein u. druckt alph. sort. Liste: Ausführ. Info: Bernhard Winkler, 5 Köln 91, Gießener Str. 42

cbm ★★ VC 64 ★★ Supersoftware zu Superpreisen über 500 Prg. ab 1 DM, max. 5 DM. Umfangreiche Liste anfordern! (Rückumschlag!) Bei Bernhard Winkler, 5 Köln 91, Gießener Str. 42

VERSCHIEDENES

VC 64/VC 20-Resettaster. Einbau ohne Lötarbeiten (einfach einstecken) DM 10,—. Maschinenprogramm für VC 64/VC 20. ReHet-Programme, die durch NEW oder Reset nicht mehr laufen DM 10,—.
Tel. 02333/80202, n. 17.00 Uhr

Tausche: Spiele, Dienst- u. Anw.-Software, sowie Simon's Basic, auf Disk.
Liste an: G. Rameil, Lannermecke 37 5940 Lennestadt 16

Tausche 1:2 und verk. über 600!! Prgr. Spiele (200) < 15 DM User < 20 DM. Bitte Liste gegen Freiumschl. Arno Beier, Königsteinerstr. 104, 4320 Hattingen, Tel. 02324/80530

★★★★ VC 20 Software ★★★★
★★ Tausch u. Verkauf 1 DM ★★
■■■■■■ Info bei: ■■■■■■
M. Reul, Scheffelstr. 32a, 5 Köln 41 ■
■ Super-Prg. auch für + 16 KByte ■

■ **MHK-CLUB:** Die richtige Adres- ■
■ se für alle die Spitzenprogr. ■
■ suchen. **Games und Utilities.** ■
■ MHK-Club: M. Maier, Osnabrük- ■
■ ker Str. 5, 8070 Ingolstadt ■

Suche gute Software aller Art (Spiele, Grafik, Geschäftspro., Textverarbeitung). Günther Seidler, Tölzer Str. 14, 1000 Berlin 33, Tel. 030/8231793

QUALITÄTSDISKETTEN mit Garantie für OEM's und Kleinhändler
5¼" ss/sd ab DM 3,44
ds/sd ab DM 3,63
mit Verst.-Ring zu 10 Stck. verp.
ADIS GmbH, Bahnhofstr. 47
8057 Eching, Tel. 089/3191205

EPROM-Platine 4 x 4 KB (8000-FFFF) DM 69 ★ Steckplatzerw. 4fach DM 128 ★ Epson-Druckerinterf., cbm-Grafik DM 298 ★ Eprommer bis 32 KB DM 298 ★ Hartge, Fohlenweide 16, 6836 Oftersheim, Tel. 06202/52675

50 Basic-Mathematik-Programme für Schule, Beruf und Freizeit programmierbar auf allen Mikro- u. Taschencomputern 20 DM. NN o. VK. Martin Kunde, Postf, 2432 Lensahn

Richten Sie Ihre Bestellungen in der Schweiz an:
Markt&Technik Vertriebs AG Alpenstr. 14 6300 Zug ☎ 042/2231 55

COMPUTER-

G. O. Hamann

Lerne BASIC mit dem Volkscomputer VC 20



Eine programmierte Unterweisung
In 24 Kapiteln auf ca. 450 Seiten werden Elemente eines Computersystems, die Phasen der Programmerstellung, die Grundlagen der Programmiersprache Basic, Programmbefehle und Systemkommandos, Farbe und Grafik, Musik und Geräusche, Sprünge und Verzweigungen, Schleifenbildung, Unterprogrammtechnik, vor- und selbstdefinierte Funktionen sowie Dateibefehle behandelt.

Bestellnummer CO 339 **DM 29,80 (Sfr. 27,50)**

Norbert Treitz

Besser Programmieren mit dem VC 20



1983, 186 Seiten
Aus dem Inhalt: Eingaben mit Input, Get, Peek, Paddles und Joystick · Editiertricks für Faulenzer · u.v.a.m.
Wenn Sie einen VC 20 haben oder kaufen wollen, ist dieses Buch eine leichtverständliche und preiswerte Ergänzung zum VC-20-Handbuch. Es enthält keine langen Programme zum Eintippen, sondern nur kurze Beispiele zum Kennenlernen der Tricks.

Bestellnummer HA 553 **DM 18,30 (Sfr. 17,10)**

K.-H. Heß

Basic-Programme für CBM/VC 20-Computer



1983, 150 Seiten
Die verschiedenen Aufgabenstellungen werden analysiert, allgemeingültige Lösungswege erarbeitet und in CBM-Basic konvertiert. Alle Programme sind ausführlich dokumentiert und anwendbar für die Serien CBM 2000, 3000, 4000 und 8000. Einige Programme laufen auch auf VC 20 und anderen basicprogrammierbaren Rechnern, wobei etwaige Programmanpassungen näher beschrieben sind.

Bestellnummer MT 501 **DM 32,— (Sfr. 29,50)**

M. Hegenbarth/M. Schäfer **NEU**

Das VC-20 Buch



1983, 351 Seiten
Dieses Buch ist eine Sammlung von gut erklärten, Programmen. Es zeigt an vielen Beispielen, daß der VC-20 längst nicht nur als Spielcomputer, sondern auch für nützliche und kommerzielle Anwendungen im kleineren Rahmen gut einsetzbar ist.
Die im Buch beschriebenen Programme sind auch auf Kassette und Diskette erhältlich.

Bestellnummer MT 516 (Buch) **DM 49,— (Sfr. 45,10)**
Bestellnummer MT 581 (Kassette) **DM 19,90 (Sfr. 19,90)**
Bestellnummer MT 582 (Diskette) **DM 29,90 (Sfr. 29,90)**

P. Rädsch **NEU**

Programme und Tips für VC-20



1983, 152 Seiten
Anhand von nützlichen und unterhaltsamen Programmen können Sie mit diesem Buch die phantastischen und selten genutzten Möglichkeiten Ihres VC-20 nun voll ausnützen. Detaillierte Beispiele zeigen, wie Sie den Befehlswortschatz Ihres Home-Computers durch einfache Routinen verbessern können. Neben Spielprogrammen finden Sie u.a. auch Programme für Textverarbeitung, Rechnungsschreibung und Lagerverwaltung.

Bestellnummer MT 513 **DM 38,— (Sfr. 35,—)**

C. Lorenz **NEU**

Beherrschen Sie Ihren Commodore 64



1983, 125 Seiten
Der Commodore 64 ist vom Konzept her gesehen ein sehr leistungsfähiges Computersystem. Warum, das werden Sie bald selbst verstehen, spätestens jedoch, wenn Sie sich eingehender mit dem C-64 beschäftigt haben. Die dazu notwendigen Ideen, Hinweise und Anregungen gibt Ihnen dieses Buch.
Neben vielen Tips und Tricks finden Sie auch Vergleiche und Hinweise auf den PET/CBM und VC-20. Dies soll es Ihnen ermöglichen, Programme aus dem Riesenvorrat von CBM-Software zu schöpfen, und diese an Ihren C-64 anzupassen.

Bestellnummer HO 533 **DM 19,80 (Sfr. 18,50)**

H. L. Schneider **NEU**

Das Commodore 64-Buch Bd. 1: Ein Leitfaden für den Erstanwender



1984, 270 Seiten
Das vorliegende Buch soll eine Unterstützung für den Erstanwender sein. Alle Möglichkeiten des Commodore 64 werden von Beginn an erklärt. Zur Einführung werden einige Programme beschrieben, die in dieser Form auch auf anderen Rechnern laufen. Erst nach dieser Grundlage wird auf die speziellen Eigenschaften des Commodore 64 eingegangen.

Bestellnummer MT 591 (Buch) **DM 48,— (Sfr. 44,20)**
Bestellnummer MT 592 (Beispiele auf Diskette) **DM 58,— (Sfr. 58,—)**

H. L. Schneider **NEU**

Das Commodore 64-Buch Bd. 2: Basic-Spiele



1984, 181 Seiten
Um Programmieren spielend zu lernen, sind in diesem Buch einige Spiele zusammengefaßt: Denkspiele, Wirtschaftsspiele, Glücksspiele und Kartenspiele. Spiele, nicht nur zum Abtippen, sondern zum selbst Ergänzen und Ändern. Alle Programme sind in der vorliegenden Version lauffähig, jedoch wurden in jedem Kapitel Anleitungen zum Ändern gegeben.

Bestellnummer MT 593 (Buch) **DM 38,— (Sfr. 35,—)**
Bestellnummer MT 594 (Beispiele auf Diskette) **DM 58,— (Sfr. 58,—)**

H.L. Schneider/W. Eberl

Das Commodore 64-Buch Bd. 3: Ein Leitfaden für Fortgeschrittene



1984, 206 Seiten
Dieses Buch ist die logische Fortführung der in Band 1 besprochenen Themen.
Einiges zu mehrfarbigen Sprites und Multi-Color-Grafik. Zum Assembler wird der Disassembler vorgestellt. Wieder viele Basic-Erweiterungen, besonders zu den Sprites (Sprites bewegen) und der hochauflösenden Grafik.

Bestellnummer MT 595 (Buch) **DM 38,— (Sfr. 35,—)**
Bestellnummer MT 596 (Beispiele auf Diskette) **DM 58,— (Sfr. 58,—)**

Tom Rugg/Phil Feldman **NEU**

Mehr als 32 BASIC-Programme für den Commodore 64



1984, 279 Seiten
Die in diesem Buch enthaltenen Programme wurden speziell für den Commodore 64 erstellt. Sie umfassen praktische Anwendungen, Lehr-/Lernhilfen, grafische Darstellungen verschiedenster Art, mathematische Aufgaben und nicht zuletzt auch einige interessante Spiele. In jedem Kapitel werden Zweck und Anwendung eines Programms erklärt, im Anschluß daran folgen ein Beispiel und das komplette Programmlisting.

Bestellnummer MT 613 (Buch) **DM 49,— (Sfr. 45,10)**
Bestellnummer MT 614 (Beispiele auf Diskette) **DM 48,— (Sfr. 48,—)**

NEU

Computerspiele und Wissenswertes — Commodore 64



1984, 156 Seiten
Dieses Buch wendet sich an alle diejenigen, die eine Sammlung von interessanten und nützlichen Maschinenprogrammen suchen. Der Leser sollte bereits etwas Erfahrung im Umgang mit Rechnern und mit der Programmierung in Maschinensprache mitbringen.
Behandelt werden alle Problemkreise, die im Mittelpunkt des Interesses stehen.

Bestellnummer MT 601 (Buch) **DM 29,80 (Sfr. 27,50)**
Bestellnummer MT 602 (Beispiele auf Diskette) **DM 38,— (Sfr. 38,—)**

Franz Ende



Das große Spielebuch — Commodore 64



1984, 141 Seiten
Das Buch soll Ihnen zeigen, welche Möglichkeiten in der Kombination von Rechnerumgebung und Software stecken. Der erste Teil enthält fertige Programme, die sofort nach dem Eintippen lauffähig sind.
Der zweite Teil des Buches wendet sich an die Leser, die etwas mehr über die Programmiertechnik erfahren wollen, mit der man die interessante Ausstattung des Commodore 64 zum Leben erweckt.

Bestellnummer MT 603 (Buch) **DM 29,80 (Sfr. 27,50)**
Bestellnummer MT 604 (Beispiele auf Diskette) **DM 38,— (Sfr. 38,—)**

Die angegebenen Preise sind die Ladenpreise. Bitte benutzen Sie die Bestellkarte auf Seite 141.

BUCHLADEN

Ihre Bestellung nehmen wir gern telefonisch entgegen:

Markt&Technik Hans-Pinsel-Straße 2, 8013 Haar bei München ☎ 089/4613-220

Günter O. Hamann

NEU

Lerne Basic mit dem Commodore 64/VC 20



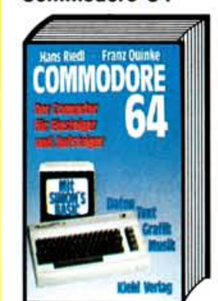
1984, 512 Seiten
Der Commodore 64 und der VC 20 haben in erheblichem Maße dazu beigetragen, daß sich heute auch der »Normalverbraucher« einen Computer leisten kann. Das Buch beschreibt die Programmiersprache dieser beiden Rechner. Nach dem Durcharbeiten aller Lektionen wird der Leser in der Lage sein, selbständig Basic-Programme für den Commodore 64, den VC 20 und die übrigen Commodore-Rechner zu erstellen.

Bestellnummer BV 652 **DM 32,80 (Sfr. 30,20)**

Hans Riedl/Franz Quinke

NEU

Commodore 64



1983, 160 Seiten
Kaum auf dem Markt, erobert sich der Mikrocomputer »Commodore 64« im Sturm die Anwendungsbereiche Arbeitsplatz, Unterricht und Freizeit.
Mit dem Commodore 64 läßt sich eine Menge machen. Doch welche Möglichkeiten Ihnen als Anwender insgesamt zur Verfügung stehen, erfahren Sie in diesem neuen Handbuch: »Commodore 64, Daten, Text, Grafik, Farbe, Musik«. Anerkannte Fachleute schreiben hier über alles, was Sie über den Commodore 64 wissen müssen.

Bestellnummer KI 617 **DM 29,90 (Sfr. 27,50)**

C. Lorenz

NEU

64 Programme für den Commodore 64



1983, 215 Seiten
Die hier beschriebenen 64 Programme bieten Ihnen alles, was Sie sich von einer solchen Programmsammlung erwarten. Viele nützliche Programme fürs Büro und viele für die Freizeit. Ebenso ist eine Zusammenstellung von Modulen enthalten, die Sie für Ihre eigenen Programme gut verwenden und dort einbauen können. Dazu gehören z.B. Eingabemaske, Rand, grafische Darstellungen usw. Für Schüler sollen die Mathematikprogramme eine Hilfe sein.

Bestellnummer HO 560 **DM 39,— (Sfr. 35,90)**

NEU

Programmieren mit dem CBM



1983, 138 Seiten
Der Großteil dieses Buches besteht aus einer umfangreichen BASIC-Programmsammlung. Hier finden Sie eine bunte Palette vom Black-Jack-Spiel über Auftragsabwicklung bis hin zum Heilkräuterprogramm.
Eine interessante und leicht verständliche Einführung in die Programmierung in 6502-Maschinensprache schließt sich an. Beispiele und ein paar Tricks machen das Experimentieren mit Ihrem CBM zu einem Vergnügen.

Bestellnummer HO 571 **DM 29,80 (Sfr. 27,50)**

CBM-Computer-Handbuch



1981, 528 Seiten
Dieses Buch beschreibt alle CBM-Computer: Den ursprünglichen CBM 2001 Personal Electronic Transactor, mit dem CBM berühmt wurde, und die danach eingeführten Serien CBM 3001, CBM 4001 und, als letztes Modell, den CBM mit einem 80spaltigen Bildschirm. Selbst wenn Sie nie an einem Computer gesessen und programmiert haben, wird Sie dieses Buch befähigen, eigene Basic-Programme für Ihren CBM-Computer zu schreiben.

Bestellnummer PW 318 **DM 59,— (Sfr. 54,30)**

Programmieren von Mikrocomputern 5 Lehr- und Übungsbuch für die Rechnerserien CBM 4001 und CBM 8001



1983, 119 Seiten
Die Rechner der Serien CBM 4001 und CBM 8001 können in CBM-Basic programmiert werden, einer Basic-Variante mit verschiedenen herstellerspezifischen Besonderheiten. Dieser Band enthält eine umfassende Einführung in CBM-Basic, ohne daß Programmierkenntnisse vorausgesetzt werden. Neben der Vermittlung der Sprache wird an verschiedenen Beispielen der Weg von der ersten Lösungsidee bis zum fertigen Programm demonstriert.

Bestellnummer VV 346 **DM 29,80 (Sfr. 27,50)**

W. Maaß

Software-Schnellkurse Multiplan, CP/M, CP/M-86, MailMerge, dBASE II, SuperCalc, WordStar, MS-DOS



1984, 85 Seiten
Was man von CP/M unbedingt kennenlernen muß · Die wichtigsten Befehle des 8-Bit-Standard-Betriebssystems und ihre Handhabung · Die wichtigsten Befehle für den täglichen Umgang.

Best.-Nr. MT 605
DM 37,- (Sfr. 34,-)



1984, 104 Seiten
Alle notwendigen Informationen für eine schnelle Einarbeitung Serienbriefe mit WordStar und MailMerge im Nu erstellt.

Best.-Nr. MT 606
DM 37,- (Sfr. 34,-)



1984, 110 Seiten
Das Datenbanksystem für Mikrocomputer kurz und bündig erklärt · Eine praktische Kurzbeschreibung für Eilige.

Best.-Nr. MT 607
DM 37,— (Sfr. 34,—)



1984, 128 Seiten
Alle Möglichkeiten von SuperCalc 2 in Kurzform · Ein Kurzüberblick über alle SuperCalc-Kommandos.

Best.-Nr. MT 608
DM 37,- (Sfr. 34,-)



1984, 88 Seiten
Was man für den Umgang mit WordStar wissen muß · Alles Wissenswerte in Kurzform · WordStar kurz und knapp erklärt.

Best.-Nr. MT 609
DM 37,- (Sfr. 34,-)



1984, 111 Seiten
Der schnelle Einstieg in Multiplan · Alle Befehle und die wichtigsten mathematischen Funktionen · Tips für den praktischen Einsatz.

Best.-Nr. MT 610
DM 37,- (Sfr. 34,-)



1984, 93 Seiten
Der tägliche Umgang mit dem Betriebssystem · Wie man Dateien anlegt, kopiert, sichert, löscht.

Best.-Nr. MT 615
DM 37,- (Sfr. 34,-)



1984, 88 Seiten
MS-DOS für den Alltag · Die Handhabung von Dateien bei Disketten und Festplatten · Für DOS 2.0 als MS- und PC-Version.

Best.-Nr. MT 651
DM 37,- (Sfr. 34,-)

Richten Sie Ihre Bestellungen in der Schweiz an:
Markt&Technik Vertriebs AG Alpenstr. 14 6300 Zug ☎ 042/22 31 55

COMPUTER-

H.P. Blomeyer-Bartenstein

Personal Computer — das intelligente Werkzeug für jedermann



1983, 352 Seiten
Dieses Buch ist der Nachfolger des Standardwerks »Personal Computer — Kompaktrechner im Einsatz«. Es faßt den aktuellen Stand der Personal Computer-Technik zusammen: Was ist und kann ein Personal Computer · Einsatzgebiete · Aufbau und Funktionsweise von Personal Computer-Systemen · Zentraleinheit · Tastatur und Bildschirm · Massenspeicher · Schnittstellen · Hardware-Erweiterungen · Mehrbenutzer-Systeme · Netzwerke · Betriebssysteme · Programmiersprachen im Vergleich · Software woher · Auswahlkriterien · Blick in die Zukunft

Bestellnummer MT 508 **DM 53,— (Sfr. 48,80)**

Personal Computer Lexikon



1982, 136 Seiten, Register: englisch-deutsch
Dieses Lexikon wurde entwickelt, um die Welt der Personal Computer transparenter zu machen. Es enthält die über 1000 wichtigsten Hard- und Software-Begriffe des »Personal Computing« und verwandter Gebiete. Alle Begriffe werden auf deutsch erklärt. Zusätzlich wird die englische Übersetzung des deutschen Suchbegriffes angegeben. Wichtig: Im Anhang befindet sich ein Register englisch-deutsch.

Bestellnummer MT 390 **DM 19,80 (Sfr. 18,50)**

D. A. Brain **NEU**

Basic-Dialekte im Vergleich



1984, 105 Seiten
Wie man Apple-, Commodore- und TRS-80-Programme untereinander konvertiert. Trotz ihrer unterschiedlichen Schreibweise löst der größte Teil der Befehle die gleichen oder ähnliche Funktionen aus. Das Ergebnis der Untersuchungen einiger verschiedener Basic-Dialekte liegt in diesem Buch vor. Es soll dem Leser helfen, Programme auf andere Programmdialekte zu übertragen.

Bestellnummer MT 564 **DM 32,— (Sfr. 29,50)**

Dr. P. Albrecht

Planen und kalkulieren mit MULTIPLAN®



Eine Einführung in das Arbeiten mit MULTIPLAN® unter CP/M-80- und MS-DOS-Betriebssystemen
1982, 225 Seiten
Eines der neuesten, leistungsfähigsten sowie gleichzeitig für den Benutzer komfortabelsten Kalkulationsprogramme ist das in diesem Buch vorgestellte MULTIPLAN.
Nach einer Einführung werden anhand von Beispielen die Befehle und Funktionen von MULTIPLAN beschrieben, und zwar in der Reihenfolge, wie sie der Arbeit in der Praxis entsprechen.

Bestellnummer MT 502 **DM 58,— (Sfr. 53,40)**

R.E. Williams

MULTIPLAN richtig eingesetzt



Alle Tricks der Tabellenkalkulation erklärt an 10 praxisnahen Beispielen
1983, 211 Seiten
Ein Übungsbuch mit Beispielen für Anwender des Computer-Programms Multiplan und solche, die es werden wollen. Wenn Sie die Übungen Schritt für Schritt durchgehen und ausprobieren, werden Sie sehr schnell in der Lage sein, die Vielzahl der Anwendungsmöglichkeiten von Multiplan zu erkennen und zu beherrschen.

Bestellnummer MT 507 (Buch) **DM 58,— (Sfr. 53,40)**
Bestellnummer MT 623 Beispiele auf Diskette (5¼", IBM-PC mit MS-DOS 2.0) **DM 48,— (Sfr. 48,—)**

J. J. Purdum

BASIC-80 und CP/M



1983, 296 Seiten
Es ist die Absicht dieses Buches, dem Leser zu zeigen, wie Mikrocomputer in Basic programmiert werden. Der Unterschied zu vielen anderen Lehrbüchern ist vor allem in zwei Punkten zu sehen:
1. Das Buch orientiert sich an einem bestimmten Basic-Dialekt: Dies erlaubt die eingehendere Behandlung von speziellen Eigenschaften.
2. Das Buch geht von einem speziellen Betriebssystem aus: CP/M. Dies wird in der sonstigen Literatur meist völlig vernachlässigt.

Bestellnummer MT 525 **DM 48,— (Sfr. 44,20)**

Rebecca Thomas/Jean Yates **NEU**

Unix-Anwenderhandbuch



1983, ca. 500 Seiten
UNIX hat sich bereits in großem Maße so bewährt, daß die Fachwelt darin das Betriebssystem der Zukunft sieht. Dieses Buch ist der richtige Leitfaden dazu. Es vermittelt Computer-Neulingen wie auch bereits Fortgeschrittenen alle Kenntnisse zum erfolgreichen Arbeiten mit diesem Betriebssystem. Das Werk ist so praxisnah erstellt, daß der Leser bereits nach kurzer Zeit die Arbeit mit seinem Computer aufnehmen kann. Der richtige Einstieg in das Betriebssystem UNIX.

Bestellnummer PW 555 **DM 79,— (Sfr. 72,70)**

Dietrich Senftleben **NEU**

Programmieren mit Logo

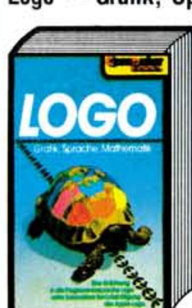


1983, 352 Seiten
Logo ist die neue Programmiersprache für Personal Computer im Ausbildungs- und Freizeitbereich. In Verbindung mit der Turtlegrafik werden neue Ideen zum mathematisch-naturwissenschaftlichen Unterricht mit Kindern verwirklicht. Logo ist jedoch keine simple Kindersprache, sondern wird sogar einem Informatikunterricht mit höchsten Ansprüchen gerecht. Es ist ein funktionsorientiertes Programmieren und unterstützt damit in idealer Form das Top-down- oder Bottom-up-Prinzip.

Bestellnummer VO 556 **DM 30,— (Sfr. 27,70)**

Herbert Stein **NEU**

Logo — Grafik, Sprache, Mathematik



1984, 257 Seiten
Logo läßt sich nicht mit den üblichen Maßstäben messen. Die Programmiersprache beinhaltet einerseits die Einfachheit des weit verbreiteten Basic, zeigt andererseits aber gleichzeitig über ein Pascal-ähnliches Prozedurkonzept Möglichkeiten auf, die eine Verwendung als Lern- und Lehrsprache sinnvoll erscheinen lassen.
Mit Logo können in kürzester Zeit grundlegende Theorien der Geometrie an allgemeinbildenden Schulen oder mathematische Problemstellungen an Universitäten dargestellt werden.

Bestellnummer MT 648 **DM 42,— (Sfr. 38,60)**

W. Pest **NEU**

Hardware-Auswahl leicht gemacht



3. völlig überarbeitete und aktualisierte Ausgabe 1984/85, ca. 400 Seiten
Die wichtigsten Daten von über 200 Personal Computer-Systemen. Mit aktuellen Marktübersichten für Personal Computer sowie die wichtigsten Peripheriegeräte, mit einführenden Artikeln zu den verschiedenen Gerätetypen, Begriffserläuterungen, Auswahlkriterien (Checklisten), Trendberichten und Bezugsquellen will das vorliegende Buch sowohl dem Computer-Einsteiger als auch dem »Profi« bei der Hardware-Auswahl eine Hilfestellung geben.

Bestellnummer MT 350 **DM 58,— (Sfr. 53,40)**

Software-Auswahl leicht gemacht



1983, 423 Seiten, 2000 Programmbeschreibungen
Dieses Buch gibt Auskunft über Systemsoftware, branchenneutrale Anwendungssoftware, branchenorientierte Anwendungssoftware und technisch-wissenschaftliche Software in Form von Kurzbeschreibungen der einzelnen Softwarepakete. Mehr als 2000 Programmbeschreibungen aus allen Anwendungsbereichen für Personal Computer.

Bestellnummer MT 340 **DM 58,— (Sfr. 53,40)**

P. Ewald

Software richtig eingekauft



1983, 144 Seiten
Informationen, Tips, Auswahlmethoden und Vorgehensweisen für alle, die sich Suche, Analyse, Leasing oder Kauf der richtigen Software erleichtern möchten.
Aus dem Inhalt: Auf der Suche nach Software · Die engere Wahl · Beginn der Entscheidung — Vertragsverhandlungen · Die endgültige Entscheidung · Stichwortverzeichnis u.a.

Bestellnummer MT 505 **DM 34,— (Sfr. 31,30)**

Die angegebenen Preise sind die Ladenpreise. Bitte benutzen Sie die Bestellkarte auf Seite 141.

Die hier vorgestellten Tips und Tricks wurden von Volker Mücke gesammelt. Wenn Sie selbst einige unbekannte Peeks und Pokes gefunden haben, so sind wir gerne bereit, diese zu veröffentlichen.

Tips und Tricks für de

Umgang mit Floppy/Drucker:

Der Drucker VC 1515 ist eigentlich nur für den VC 20 gedacht. Will man ihn für den C64 benutzen, so funktioniert dies nicht ohne weiteres. Dazu benötigt man zwei spezielle POKE-Befehle:

Vor dem Ansprechen des Druckers muß man POKE 53265,11 und nach Beendigung des Druckvorgangs POKE 53265,27 eingeben. Nachteilig ist allerdings, daß während des Druckens die Schrift auf dem Bildschirm verschwindet.

Das gleiche Problem ergibt sich beim Gebrauch der Floppy VC 1540 zusammen mit dem C 64. Auch hier braucht man die beiden POKE-Befehle. Allerdings müssen sie anders gehandhabt werden: Man gibt, wenn man laden will, folgendes im Direktmodus ein:

POKE 53265,11:POKE 53266,27:LOAD "NAME",8 (bzw.8,1)

So, und nun lädt der C 64 auch von der Floppy VC1540.

Joystickabfrage:

Auch die Joystickabfrage ist beim C 64 nicht ganz einfach. Dabei hilft folgendes Hilfsprogramm:

```
10  POKE 56322,224
20  JO=PEEK(56320)
30  IF (JO AND 1)=0 THENPRINT"OBEN"
40  IF (JO AND 2)=0 THENPRINT"UNTEN"
50  IF (JO AND 4)=0 THENPRINT"LINKS"
60  IF (JO AND 8)=0 THENPRINT"RECHTS"
70  IF (JO AND 16)=0 THENPRINT "FEUER"
```

Dieses Programm gilt für eine Joystickabfrage am Controlport 2.

Die Funktion FRE(X):

Um bei der Abfrage PRINT FRE(X) die richtige Ausgabe zu erhalten, muß man folgendes noch hinzufügen: PRINT FRE(X)+65538.

Nun gibt der Computer die tatsächlichen freien Bytes aus.

Listschutz

Wenn man ein Programm hat, welches irgendeinen Listschutz hat, so will der Benutzer doch versuchen, dieses Programm zu »knacken«. Wenn zum Beispiel die Commodore-Taste ausgeschaltet wurde, so kann man beim C 64 — anders als beim VC 20 — die Control-Taste benutzen, zum Beispiel:

CTRL + "N" = Kleinschrift
CTRL + "M" = RETURN
CTRL + "S" = HOME
CTRL + ";" = Cursor rechts
CTRL + "=" = Cursor blinkt
CTRL + "Q" = Cursor unten
CTRL + "T" = Cursor links

Reset-Auslöser:

SYS 64738 bewirkt beim C 64 einen System-Reset.

Einige Peeks und Pokes

?ST oder ?PEEK(144) :Statusabfrage

POKE 207,X [1< =X< =255]: Bestimmung der Schnelligkeit der Blinkphase des Cursors.

PRINT PEEK(214): gibt die Zeile an, in der der Cursor gerade ist.

PRINT PEEK(202): gibt die Spalte an, in der der Cursor im Augenblick ist.

POKE 214,X [1< =X< =23]: versetzt den Cursor an die Zeile x

PRINT PEEK(647): gibt die Farbe unter dem Cursor an.

POKE 647,X: Farbenwechsel unter dem Cursor.

PRINT PEEK(215): gibt die letzte gedrückte Taste im Characterwert aus

PRINT PEEK(213): gibt die Länge der momentanen Zeile auf dem Bildschirm an.

Rund um den OUTPUT:

PRINT PEEK (183): Ausgabe der Länge des Filenamens.

PRINT PEEK (184): Ausgabe der laufenden Filenummer.

PRINT PEEK (185): gibt die laufende Sekundäradresse an.

PRINT PEEK (186): gibt die laufende Primäradresse an.

Weitere nützliche Peeks und Pokes

POKE 652,X [1< =X< =255]: verzögert die Repeat-Funktion

SYS 65499: Der Systembefehl bewirkt, daß TI$ auf 0 gesetzt wird.

PRINT PEEK (36870): Ausgabe der Lichtgriffel Horizontal-Position.

en Commodore 64

PRINT PEEK (36871): Ausgabe der Lichtgriffel Vertikal-
Position.
0 POKE 199,1:PRINT"Volker Muecke": Diese Zeile be-
wirkt, daß die PRINT-Anweisung den Text invers aus-
gibt.
0 POKE 199,0:PRINT"Volker Muecke": dagegen gibt
den Text wieder normal aus.
PRINTPEEK(10) oder PEEK
(147): Ist dieser Wert = 0, so wurde als letzter Device-
Befehl ein LOAD eingegeben, wenn der Wert = 1 ist,
wurde VERIFY eingegeben.

Zerstörungsbefehle

Code-Wörter sind — gegen Unbefugte — sehr nützlich.
Allerdings muß auch bei Falsch-Eingabe die geeignete
Reaktion vom Computer aus folgen.

Hier ein Beispiel:

```
10  INPUT "Code-Wort" ??";A$
20  IF A$="Volker Muecke"THENPRINT"Programm
      zugänglich !":END
30  POKE 776,1
```

Erklärung: In Zeile 10 wird der Anwender gebeten,
das Code-Wort einzugeben. Ist die Überprüfung richtig,
so kann er mit dem Programm arbeiten. Stimmt die Ein-
gabe nicht mit dem Code-Wort überein, so wird das Pro-
gramm in Zeile 30 zerstört.

Nun könnte man natürlich das Programm einfach aufli-
sten und nachsehen, welches das richtige Code-Wort ist.
Das heißt, man müßte versuchen, das Code-Wort und
den Zerstörungsbefehl »unsichtbar« zu machen, so daß
er beim Auflisten nicht mehr zu sehen ist. Dazu hilft fol-
gendes Zusatzprogramm, welches in das bestehende
Programm eingefügt wird:

```
60000 H=44:L=43:X=PEEK(H)*256+PEEK(L):INPUT
        "Zeilennummer:";Z
60020 Y=PEEK(X+1)*256+PEEK(X)
60030 IF PEEK (X+3)*256+PEEK(X+2)=Z THEN
        POKE X+4,0:PRINT "OK-OK":END
60040 IF PEEK (X+1)*256+PEEK(X)=0 THEN
        PRINT"Zeile nicht gefunden !":END
60050 X=Y: GOTO 60020
```

Bevor man dieses Unterprogramm nun starten kann,
muß man folgendes machen: Man setzt nach der Zeilen-
nummer der Zeile, die verschwinden soll, fünf '+'-Zei-
chen.

In unserem Fall wäre das:

20 +++++IF A$="Volker Muecke"....und so weiter
und
30 +++++POKE 776,1

So, nun startet man das Unterprogramm mit RUN
60000. Dieses Programm macht alles nach den ge-
wünschten Zeilennummern unsichtbar, so daß da steht:

20
30

obwohl das Programm noch korrekt läuft.

Hier noch einige Zerstörungsbefehle und ihre Auswir-
kungen:

POKE 776,1:
Dieser Befehl bewirkt, daß das ganze Programm mit un-
definierbaren Zeichen vollgeschrieben wird.

POKE 777,1:
Hier wird kein Befehl mehr ausgeführt. Der Cursor
springt jeweils in die Ecke links oben.

POKE 770,0:
Das ist auch ein sehr schöner Zerstörungsbefehl. Er be-
wirkt, daß eine READY-Ausgabe unendlich oft erfolgt.
Will man nun das Programm laufen lassen, besteht keine
weitere Möglichkeit mehr, als den Computer auszuschal-
ten.

Hat man aber nach einiger Zeit sein eigenes Code-
Wort vergessen, so besteht auch noch eine Möglichkeit,
dieses Code-Wort zu finden, indem man die unsichtba-
ren Zeilen eben wieder sichtbar machen muß. Dies ge-
schieht mit folgender Routine, die an das eigentliche Pro-
gramm angehängt wird:

```
60000  AN=PEEK(43)+PEEK(44)*256
60010  P=PEEK(AN)+256*PEEK(AN+1)
60020  IF P=0 THEN PRINT "Fertig!":END
60030  IF PEEK (P+4) THEN 60050
60040  POKE P+4,32:POKE P+5,32:POKE P+6,32:PO-
         KE P+7,32:POKE P+8,32
60050  P=PEEK(P)+256*PEEK(P+1):GOTO 60020
```

Selbstverständlich soll man nach jeder »Operation«
den Codierer beziehungsweise Decodierer aus dem
Hauptprogramm nehmen, damit niemand dies dazu aus-
nutzen könnte, das Programm zu »knacken«. Viel Spaß
beim Decodieren!

In der nächsten Ausgabe gibt es weitere PEEK und
POKES.

Teil 1

Strubs — ein Precompiler für Basic-Programme

In dieser und den folgenden Ausgaben des 64'er wollen wir Ihnen ein Programm für den Commodore 64 und VC 20 mit dem Namen »Strubs« — das steht für »strukturiertes Basic« — vorstellen. Es handelt sich bei dem Programm um einen sogenannten Precompiler, ein Programm, welches Programmtexte mit gewissen zusätzlichen Befehlen in normale, auf jedem Commodore 64 oder VC 20 ablauffähigen Basic-Programm übersetzt.

In der ersten Folge wollen wir zunächst beschreiben, was solch ein Precompiler (= Vorübersetzer) eigentlich macht und dessen Methode mit der Arbeitsweise von Interpretern auf der einen und Compilern auf der anderen Seite vergleichen. Anschließend wird dann das Programm »Strubs« kurz vorgestellt. Weitere Teile werden die Grundlagen der strukturierten Programmierung und die praktische Entwicklung von eigenen Programmen mit Hilfe von Strubs behandeln. Zu guter Letzt ist noch ein Teil vorgesehen, in welchem genauer auf den Aufbau des Programms, auf Änderungs- und Erweiterungsmöglichkeiten eingegangen wird. Das Programm Strubs wurde ursprünglich zu einer Zeit entwickelt, als Begriffe wie Forth oder Pascal noch Fremdworte für den C 64 waren. Der Zweck war die Entwicklung von Programmen übersichtlicher, effizienter und bequemer zu gestalten.

Strubs bietet neue Basicbefehle

Auf der einen Seite ermöglicht es Strubs, auf sanftem Weg, das heißt im Rahmen des gewohnten Basic (aber ohne auf unübersichtliche Klimmzüge innerhalb des Commodore Basic angewiesen zu sein), also ohne gleich eine neue Programmiersprache lernen zu müssen, mit der Technik strukturierter Programmierung vertraut zu werden. Auf der anderen Seite ermöglicht es Strubs, sich mit der Arbeit mit Compilern vertraut zu werden.

Schließlich bietet die Form des Precompilers noch erhebliche Geschwindigkeitsvorteile gegenüber vergleichbaren Interpretererweiterungen. Um diese letzten beiden Punkte zu verstehen, ist es angebracht, auf die unterschiedlichen Arbeitsweisen von Interpretern und Compilern einzugehen.

Bekanntlich versteht der eigentliche Computer, das heißt hier der Mikroprozessor, nur die sogenannte Maschinensprache. Da diese aber extrem problemfern und unübersichtlich ist, hat man verschiedene höhere Programmiersprachen erfunden, um dem Programmierer seine Arbeit zu erleichtern. Damit aber ein in einer solchen Sprache geschriebenes Programm vom Computer verarbeitet werden kann, muß zunächst eine Übertragung in die Maschinensprache des Computers stattfinden. Dabei wird diese Übertragung wiederum von Programmen vorgenommen und zwar von Compilern oder von Interpretern. Diese beiden Programmarten unterscheiden sich grundlegend in ihrer Arbeitsweise.

Ein Interpreter besteht im wesentlichen aus einer Reihe von in Maschinensprache geschriebenen Unterprogrammen, einer Tabelle welche die erlaubten Befehle und die Adresse des zu jedem Befehl gehörenden Unterprogramms enthält, schließlich der Variablenverwaltung sowie der sogenannten Interpreterschleife.

Diese Schleife geht den Programmtext Schritt für Schritt durch. Zu jedem Befehl sucht sie in der Tabelle die zugehörige Unterprogrammadresse, ruft dieses Unterprogramm auf, holt den nächsten Befehl und so weiter, bis das Programmende erreicht ist. Man sieht also, daß ein großer Teil der Arbeit eines Interpreters im Suchen besteht: Suchen in der Befehlstabelle, Suchen in der Variablentabelle und nicht zuletzt Suchen nach Sprungzielen im, zu interpretierenden, Programm.

Diese ewige Sucherei führt nun dazu, daß Programme nur relativ langsam abgearbeitet werden. Eine Interpretererweiterung (wie etwa Siemens Basic) stellt nun einfach zusätzliche Befehlsroutinen zur Verfügung und erweitert die Befehlstabelle um die neuen Befehle und Adressen. Durch diese Erweiterung der Befehlstabellen wird jetzt aber leider auch der Zeitaufwand für das Suchen größer, so daß die Programme noch langsamer als bisher schon ablaufen. Simons Basic demonstriert dies sehr anschaulich. Ein Beispiel für eine Interpretererweiterung werden wir weiter unten besprechen.

Nehmen wir zur Illustration der Arbeitsweise eines Interpreters eine Programmzeile wie die folgende:

FOR I = 0 to 999: PRINT I: NEXT

Der Interpreter muß hier 1000-mal die Befehlstabellen nach dem Befehl PRINT und 1000mal die Variablentabelle nach der Variablen I durchsuchen.

Compiler kontra Interpreter

Völlig anders arbeitet der Compiler: Er übersetzt ein Programm, das in einer Sprache geschrieben ist, welche nur der Programmierer versteht — dieses Programm nennt man Quellprogramm — in ein äquivalentes Programm — das Objektprogramm —, das (meist nur noch) die Maschine versteht. Diese beiden Begriffe — Quellprogramm und Objektprogramm — sollten wir uns gut merken; sie werden noch öfter auftauchen.

Der größte Teil der Sucharbeit

kann nun ein für allemal bei der Übersetzung vom Compiler geleistet werden. Die benötigten Adressen der Befehlsroutinen, der Variablen und der Sprungziele sind für immer fest in das Objektprogramm eingebaut. Dadurch können compilierte Programme oft bis zu zehnoder mehrmal schneller sein als entsprechende Interpreterprogramme.

Diesem beträchtlichen Gewinn an Geschwindigkeit steht allerdings ein nicht minder bedeutender Verlust an Bequemlichkeit gegenüber. Zum einen erfordert selbst die geringste Programmänderung eine vollständige Neuübersetzung des Programms. Dies allein kann bei umfangreichen Programmen erhebliche Zeit beanspruchen, zumal häufig auch noch diverse Zwischenschritte erforderlich sind, auf die wir hier nicht näher eingehen wollen. Zum anderen stellt das von einem einfachen Compiler erzeugte Objektprogramm für den Programmierer meist einen großen schwarzen Kasten dar, in den hineinzusehen ihm verwehrt bleibt. Er kann das Programm in der Regel nicht einfach unterbrechen, um sich bestimmte Variablenwerte anzusehen oder Variablen bestimmte Testwerte zuweisen, um damit dann einen kritischen Programmteil ausführen zu lassen, mal eben eine Zeile ändern und was der Annehmlichkeiten beim Programmtest mit einem Interpreter mehr sind. Bessere Compiler bieten zwar eine Reihe von Optionen und Hilfsprogrammen für die Fehlersuche und das Programmtesten an, jedoch bleibt auch hier, verglichen mit einem Interpreter, diese Arbeit reichlich unbequem. Ideal ist es sicherlich, äquivalente Interpreter und Compiler zur Verfügung zu haben. Auch gewisse Mischformen wie zum Beispiel bei der Sprache Forth sind hier interessant.

Strubs — eine Mischung von Interpreter und Compiler

Um nun aber auf das Programm Strubs zurückzukommen: Auch hier haben wir es in gewisser Hinsicht mit einer Mischform zu tun. Das selbst nicht lauffähige Quellprogramm, welches der Programmierer unter Benutzung der neuen Befehle erstellt, wird von Strubs nicht in Maschinensprache übersetzt, sondern in ein normales Basic-Programm, das dann wie bisher interpretiert wird. Dabei werden Programmteile, die keine Erweiterungen enthalten, mehr oder weniger unverändert übernommen. Dieses von Strubs erzeugte Objektprogramm kann nun wie jedes andere Basic-Programm — auch mit Hilfe von Toolkits — gelistet, ausgetestet und sogar geändert werden. Schließlich ist es dann noch möglich, dieses Objektprogramm mit Hilfe eines Basic-Compilers, wie zum Beispiel dem Austro Compiler, weiter zu übersetzen. Besonders hilfreich ist es, daß einander entsprechende Programmzeilen im Quellprogramm und im Objektprogramm gleiche Zeilennummern besitzen, so daß der Programmierer sich ohne Schwierigkeiten im Objektraum zurechtfinden kann. Gegenüber der Methode, den Basic-Interpreter zu erweitern, bietet dieses Verfahren Geschwindigkeitsvorteile: Diese ergeben sich einerseits aus der Tatsache, daß alle Kommentare und Leerzeichen gelöscht werden können, andererseits wird wie beim Compiler ein Teil der Sucharbeit während der Übersetzung erledigt. Dadurch werden zum Teil erst neue Anweisungen ermöglicht, deren Realisierung im Rahmen einer Interpretererweiterung zu aufwendig wäre.

Schon durch die Suche nach Sprungzielen wirkt der Basic-Interpreter langsam genug: Bei jedem Sprung wird das Programm von Anfang an durchsucht, bis die entsprechende Zeilennummer gefunden ist. Deshalb empfiehlt es sich auch, häufig aufgerufene Unterprogramme möglichst an den Programmanfang zu setzen, da sie dann schneller gefunden werden.

Wer sucht, der findet: aber wann?

Daß sich die Suchzeit überhaupt in erträglichen Grenzen hält, liegt nun daran, daß der Programmtext selbst nicht durchsucht werden muß. Vielmehr braucht der Interpreter nur entlang der Kette aus Zeilennummern und Zeigern zur nächsten Zeile zu suchen, bis die gewünschte Zeilennummer gefunden ist (Bild 1). Sollte nun der Interpreter aber bei nicht erfüllter Bedingung in einer IF-Anweisung das zugehörige ELSE suchen, bei nicht erfüllter Eingangsbedingung einer

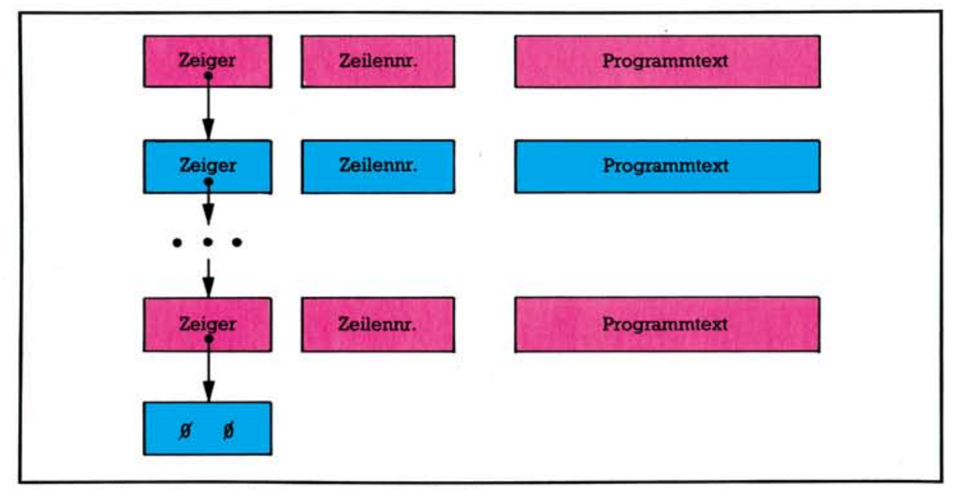


Bild 1. Aufbau eines Basic-Programms im Speicher

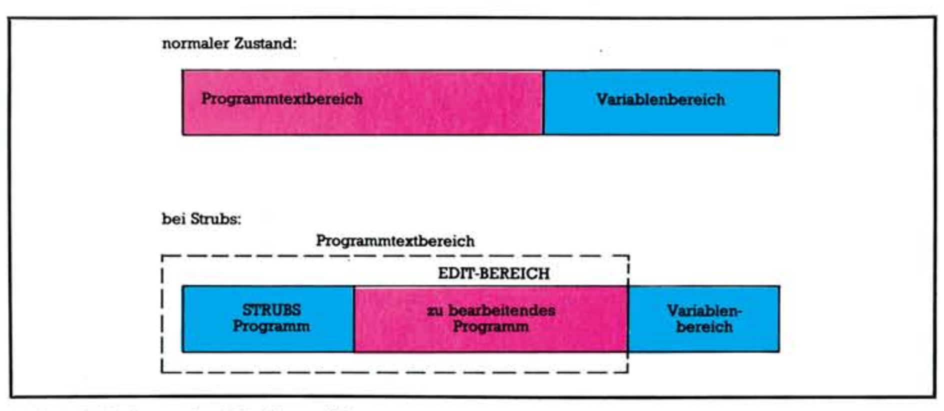


Bild 2. Aufteilung des Arbeitsspeichers

FOR-Schleife das zugehörige NEXT oder zu einem WHILE das END-WHILE, dann müßte der gesamte Programmtext selbst durchsucht werden.

Interpreter durchlaufen jede Schleife mindestens einmal

Deshalb arbeiten die Basic-Interpreter im allgemeinen so, daß solche Blöcke — wie die FOR-Schleife — mindestens einmal durchlaufen werden. Deshalb muß bei solchen Interpretern — sofern sie überhaupt ein ELSE kennen — dieses in der gleichen Programmzeile wie das zugehörige IF stehen. Deshalb kennt zum Beispiel Simons Basic die REPEAT-UNTIL-Anweisung, die immer mindestens einmal durchlaufen wird, nicht aber die WHILE-Anweisung. Ein Precompiler aber kann bei der Übersetzung den Abschlußbefehlen eines Blockes — wie ELSE oder END-WHILE — ihre Zeilennummern zuordnen, so daß beim Programmlauf nicht mehr der Programmtext selbst, sondern nur die Kette der Zeilennummern durchsucht werden muß.

Vorübersetzung nicht nur beim Precompiler

Die Methode der Vorübersetzung zur Erhöhung der Laufgeschwindigkeit benutzt im übrigen auch der Basic-Interpreter des Commodore 64. Und zwar findet sich die Übersetzungsfunktion im Editor: Sofort bei der Eingabe einer Zeile werden die Basic-Befehle, die aus mehreren Zeichen bestehen, in nur 1 Byte lange Zeichen, die sogenannten TOKENS, übersetzt. Eine Liste dieser Tokens findet sich zum Beispiel im Programmierhandbuch zum VC 20. Diese Vorübersetzung bringt zwar einen schönen Gewinn an Geschwindigkeit, hat allerdings den Nachteil, daß Programmtexte nicht mehr mit komfortableren Editor- beziehungsweise Textprogrammen erstellt werden können. Für uns ist jedoch vor allen Dingen wichtig, daß diese TOKENS berücksichtigt werden müssen, falls der Befehlsvorrat von Strubs erweitert werden soll, oder falls Programme für Interpretererweiterungen wie Simons Basic bearbeitet werden sollen. Aber auf diesen Punkt werden wir ein anderes Mal ausführlicher eingehen.

Wenn wir mit Strubs arbeiten, haben wir es — wie bei jedem Compiler — mit (mindestens) drei Programmen zu tun: Dem Übersetzungsprogramm, dem Quellcode (Quellprogramm) und dem lauffähigen Objektprogramm. Diese Programme müssen sich nun irgendwie den zur Verfügung stehenden Speicherplatz teilen. Daß das Übersetzungsprogramm, um arbeiten zu können, im Hauptspeicher stehen muß, versteht sich von selbst.

Eine Möglichkeit wäre nun, daß das Übersetzungsprogramm das Quellprogramm von der Diskette einliest, und gleichzeitig das erzeugte Objektprogramm auf Diskette schreibt. Der Compiler zu Simons Basic arbeitet zum Beispiel nach dieser Methode. Da ein Compiler jedoch einen Programmtext in der Regel mindestens zweimal durchliest — man spricht in diesem Fall von 2-Pass-Compilern —, ist es günstiger, wenn das Quellprogramm sich ebenfalls im Hauptspeicher befindet. Diesen Weg gehen zum Beispiel Pascal 64 und Strubs. Um nun den zur Verfügung stehenden Platz aufzuteilen, benutzt zum Beispiel Pascal 64 eine sehr einfache und wirksame Methode: Der Compiler ist selbst in Basic geschrieben und enthält eine unsichtbare Zeile mit der Zeilennummer 0, die ihrerseits einen Sprung zum Übersetzungsprogramm enthält. Ein Pascalprogramm wird nun einfach mit den Zeilennummern zwischen 1 und 9999 in das Compilerprogramm eingefügt.

Dieses Verfahren hat allerdings den Nachteil, daß Programme immer nur zusammen mit dem Compiler abgespeichert und editiert werden können. Insbesondere ist es damit nicht möglich, Quellprogramme aus fertigen Bausteinen (Modulen) zusammenzusetzen.

Strubs geht andere Wege

Aus diesem Grund wurde für Strubs ein anderer Weg gewählt: Entsprechend Bild 2 wurde der Speicher des Commodore 64 in drei Bereiche aufgeteilt. Am Anfang des Arbeitsspeichers steht das Programm Strubs (Pointer in Zelle 43/44). Der Edit-Bereich für Quellprogramme beginnt bei (Wert der Variablen EA). Daran anschließend befindet sich der (gemeinsame) Variablenbereich (Pointer in Zelle 45/46). Um nun vom Edit-Bereich aus bequem in den anderen Speicherbereich umschalten und die Übersetzung starten zu können, benutzt Strubs selbst eine kleine Interpretererweiterung, die, wie versprochen, kurz vorgestellt werden soll.

Die Eingabe von »!« bewirkt nun dasselbe wie die Befehlsfolge »POKE 44,8: RUN«. Das entsprechende Assemblerlisting findet sich in Bild 3. Das kleine Programm »Erweiterung« holt zunächst den nächsten Befehl. Dann muß für die Routine »Befehl ausführen« der Status gerettet werden, da die CHARGET-Rou-

```
ERWEITERUNG:
02C0    207300    JSR 0073    ; Charget, nächstes Zeichen holen
02C3    08        PHP         ; Status retten
02C4    C921      CMP _21     ; »!«, neuer Befehl?
02C6    F004      BEQ 02CC
02C8    28        PLP         ; nein, dann Status wiederherstellen
02C9    4CE7A7    JMP A7E7    ; und normalen Befehl ausführen
02CC    28        PLP         ;
02CD    A908      LDA _08     ; Erweiterungsroutine:
02CF    852C      STA 2C      ; entspricht Poke 44,8: RUN
02D1    A98A      LDA _8A     ; RUN-TOKEN
02D3    4CE7A7    JMP A7E7    ; Befehl ausführen

INIT:
02EE    A9C0      LDA _C0     ; Erweiterung, Low Byte
02F0    8D0803    STA 0308
02F3    A902      LDA _02     ; Erweiterung, High Byte
02F5    8D0903    STA 0309
02F8    60        RTS
```

Bild 3. Interpretererweiterung

tine damit wichtige Informationen übermittelt. (Dies ist wichtig und wurde in dem unten erwähnten Buch übersehen.) Nachdem verglichen wurde, ob ein neuer Befehl vorliegt, wird dann entsprechend zum normalen Programmverlauf oder zur Erweiterungsroutine verzweigt. Für eigene Versuche mit Interpretererweiterungen können an dieser Stelle beliebige Maschinenprogramme (gegebenenfalls mit weiteren Decodierungen) gesetzt werden. Nur sollte zum Abschluß — anders als hier, wo ein Basic-Befehl aufgerufen wird — ein Sprung zur Interpreterroutine ² A7E4 erfolgen, wo dann der nächste Befehl bearbeitet wird. Um nun die Erweiterung in den Basic-Interpreter einzubinden, benötigen wir dann nur noch eine kurze Initialisierungsroutine, die den Zeiger in ² 0308 auf den Anfang der Erweiterung setzt.

Wer selbst solche Erweiterungen entwickeln möchte, findet weitere Informationen und viele Anregungen in dem Buch »64 Intern« von Data Becker. Für weitergehend Interessierte empfiehlt sich die gut verständliche Einführung »Compilerbau« von N. Wirth, Teubner, Stuttgart 1981.

Strubs stellt sich vor

Abschließend wollen wir nun das Programm Strubs kurz vorstellen. Am Anfang der Programmentwicklung standen folgende Vorstellungen, die durch das Programm erfüllt werden sollten:

1. Unabhängigkeit von Zeilennummern
2. Unterstützung strukturierter Programmierung
3. Unterstützung modularer Programmentwicklung
4. Erweiterung der Dokumentationsfähigkeit des Programmtextes. Dabei sollte das Programm
5. einfache Handhabung gewährleisten und
6. effiziente Fehlersuche ermöglichen.

Die Unabhängigkeit von Zeilennummern wird erreicht durch die Verwendung beliebig langer Labels oder relativer Sprünge anstelle von Zeilennummern.

Die wichtigsten Kontrollstrukturen höherer Programmiersprachen werden von Strubs zur Verfügung gestellt:

IF — THEN — FI
IF — THEN — ELSE — FI
WHILE — EWHILE
REPEAT — UNTIL
LOOP — EXIT (beliebig oft) — ELOOP
CASEOF — OF (beliebig oft) — ELSE (optional) — ECASE

Durch die Unabhängigkeit von Zeilennummern und eine EXTERN-DECLARATION wird das Anlegen einer Modulbibliothek — sowohl auf Quellprogramm- als auch auf Objektprogrammebene — unterstützt.

Mit Strubs werden Sie ein vielseitiges Werkzeug in Händen halten

Der Dokumentationsfähigkeit des Programmtextes dienen neben den bereits erwähnten Marken und Kontrollstrukturen ein Tabulator und Kommentare an beliebiger Stelle auch innerhalb einer Zeile, ja selbst innerhalb eines Variablennamens (zum Beispiel A'US'G'ABE'% = AG%).

Programmtexte können wie gewohnt mit dem normalen Basic-Editor geschrieben werden.

Schließlich werden wir zur Illustration der Erweiterung des Befehlssatzes von Strubs noch eine MAKRO-Funktion implementieren. Von besonderer Bedeutung ist, daß das Programm von Anfang an unter dem Aspekt möglichst einfacher Erweiterbarkeit konzipiert wurde. Damit konnte das Programm im Bootstrapping-Verfahren entwikkelt werden, so daß es jetzt selbst sowohl als Quellprogramm als auch als Objektprogramm vorliegt. Wem es Spaß macht, der mag Strubs einfach auch als ein generelles Übersetzungsprogramm zur Aufbereitung von Programmtexten auffassen und seine gegenwärtigen Features als Beispiel möglicher Implementationen. In der nächsten Ausgabe werden wir das komplette Objektprogramm von Strubs abdrucken. (Matthias Törk)



Alle Tasten-, Zeichen- und Steuercodes

Haben Sie schon einmal einen Ball länger als ein paar Minuten auf derselben Stelle liegen sehen? Ich noch nicht, denn der nächste Vorbeikommer kickt ihn — irgendwohin. Sie doch auch, oder? Dasselbe Phänomen können Sie überall dort beobachten, wo ein Heimcomputer steht. Eingeschaltet oder nicht — jeder, der vorbeikommt, drückt auf eine Taste. Und wenn das Ding gar reagiert — mit einem Buchstaben auf dem Bildschirm — dann bleibt selbst die Oma stehen und tippt nochmal und nochmal.

Die Tasten wirken auf die Menschen wie das Licht auf die Motten

Wenn vom VC 20 oder Commodore 64 die Rede ist, kommt dieselbe sehr rasch auf die Tastatur. Ihre Schreibmaschinen-Qualität und die klaren Steuer- und Funktionstasten heben sie aus der Schar der Konkurrenten heraus. Sie ist sehr benutzerfreundlich, besonders für Anfänger, während der ersten Tast(en)versuche.

Aber rasch wird offensichtlich, daß die Tasten auch ihre Geheimnisse haben, und zwar frühestens, wenn die vier Funktionstasten keine Reaktionen, zumindest keine sichtbaren, hervorrufen und spätestens bei dem Versuch, in einem Programmablauf bestimmte Tastenfunktionen ausführen zu lassen, wie zum Beispiel »Cursor Down« oder »Bildschirm löschen«.

Ich will Ihnen deshalb in diesem Aufsatz die Grundlagen der Tastaturabfrage, das Geheimnis der Funktionstasten, die CHR$-Anhängsel von PRINT-Befehlen, die mysteriösen »negativen« Zeichen in Anführungsstrichen und gleichzeitig gedrückte Tasten erklären und mit Kochrezepten ausschmücken.

Übrigens: das hier Gesagte gilt sowohl für den VC 20 als auch für den Commodore 64. Nur die Programmbeispiele nicht, die sind auf den kleinen VC 20 zugeschnitten. Die 64er mögen mir das verzeihen. Ich scheue mich aber, Kochrezepte abzugeben, die ich nicht selbst ausprobiert habe. Und ich habe leider (noch) keinen C 64. Falls Unterschiede bei den Adressen auftreten, gebe ich den Wert für den C 64 in Klammern an.

Als allererstes möchte ich folgende Behauptung aufstellen:

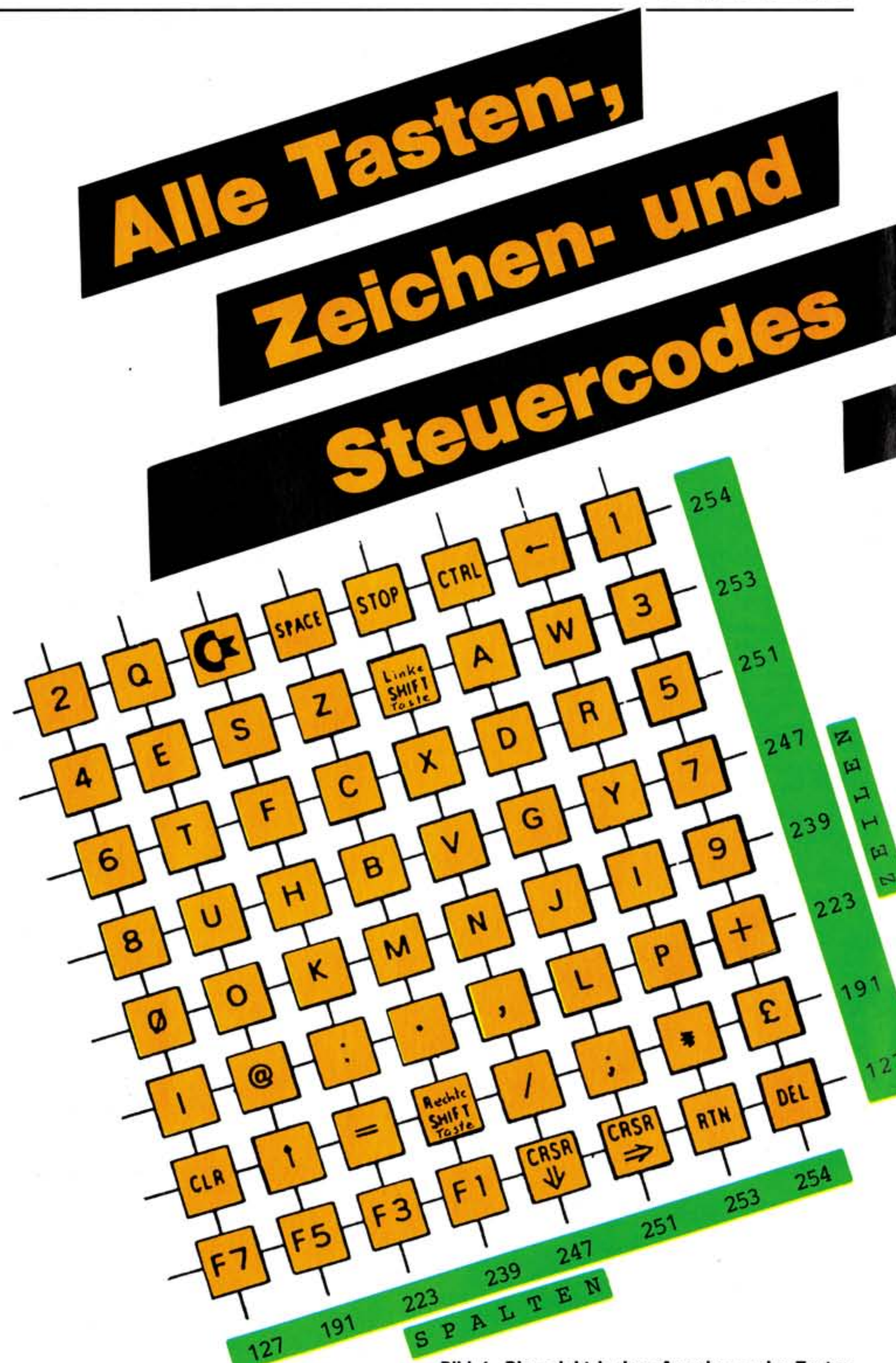


Bild 1. Die »elektrische« Anordnung der Tasten

Dem Computer sind alle Tasten gleich

Und er behandelt sie auch alle gleich. 60 mal in jeder Sekunde unterbricht der Computer was er auch immer gerade ausführt, merkt sich, wobei und wo er gerade unterbrochen hat, und schaut nach, ob eine der Tasten gedrückt worden ist.

Keine Regel ohne Ausnahme, auch beim Computern nicht: Die RESTORE-Taste und die SHIFT-LOCK-Taste fallen völlig aus dem Rahmen und haben mit den kommenden Erklärungen nichts zu tun. Ich werde sie deshalb auch nicht mehr erwähnen. Der Vollständigkeit halber sei hier nur kurz gesagt, daß die RESTORE-Taste den Computer bei jeglicher Arbeit unterbricht und ihn mit READY und blinkendem Cursor in den Anfangszustand zurücksetzt. Die SHIFT-LOCK-Taste ist (wie bei der Schreibmaschine) eine mechanische Arretierung der SHIFT-Taste.

Es bleiben uns immerhin 64 Tasten, die vom Computer während seiner Verschnaufpause inspiziert werden. Die Funktionstasten sind

(Teil 1)

Die gute und übersichtliche Tastatur der Computer von Commodore bietet gerade dem Anfänger sehr leichte Einstiegsmöglichkeiten. Wer aber tiefer in die Computerei eindringt, stößt beim VC 20 und Commodore 64 schnell auf eine verwirrende Vielfalt von Codes und Steuerzeichen, deren Kenntnis und Beherrschung jedoch auch dem Anfänger ein weites Anwendungsfeld eröffnen kann.

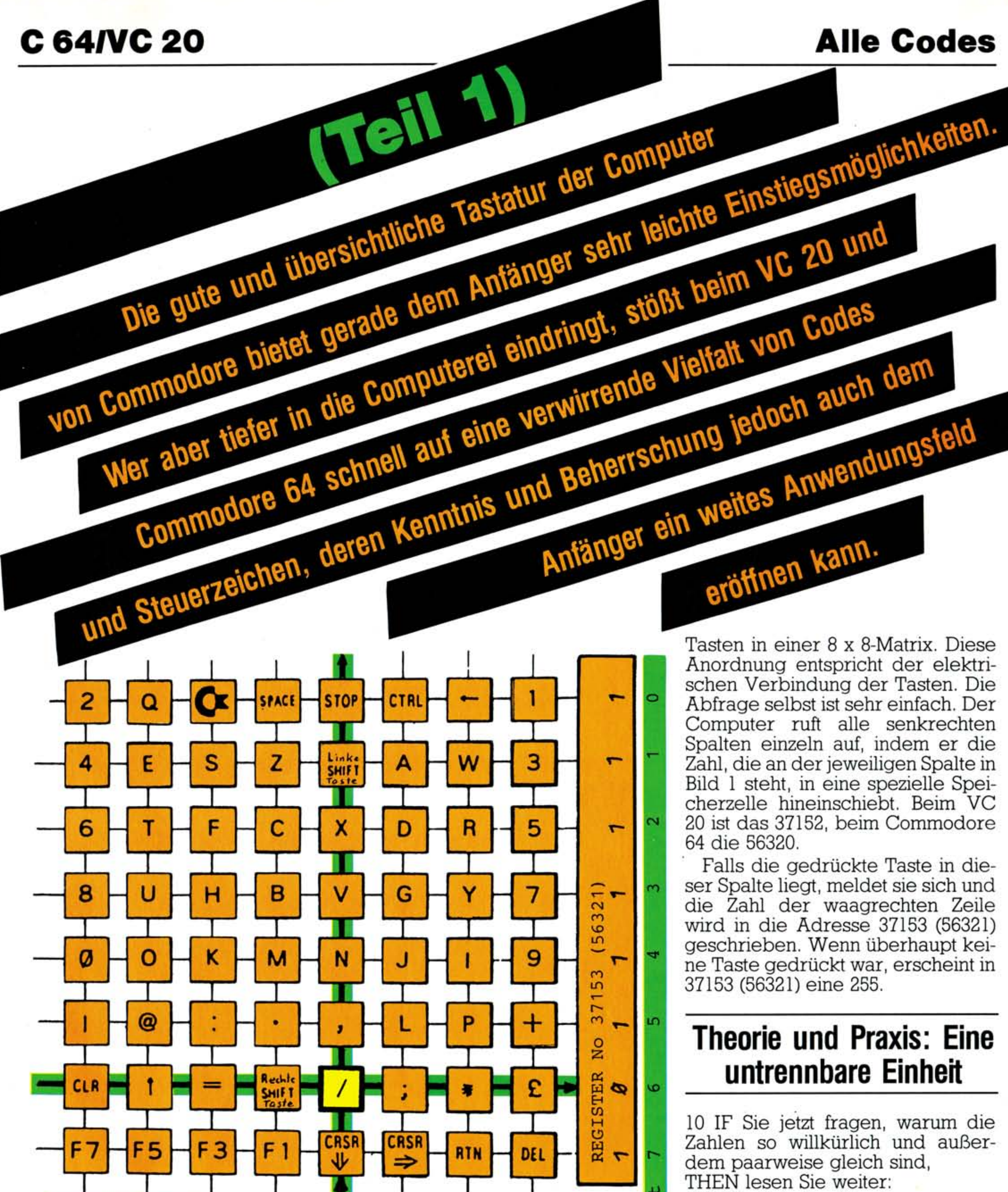


Bild 2. Die Tasten-Matrix mit ihren Registern

immer dabei! Diese Inspektion — Tastaturabfrage genannt — wollen wir uns näher anschauen.

Der Computer erhält nicht, wie es eigentlich logisch wäre, von jeder gedrückten Taste ein spezielles Code-Signal. Das wäre für einen Homecomputer zu aufwendig und zu teuer. Das Betriebssystem des Computers veranstaltet vielmehr eine Befragung seines Tastenvolkes, nach dessen Stimmenabgabe er dann entscheidet, welche Taste nun eigentlich gedrückt worden ist.

Schauen Sie sich bitte das Bild 1 genauer an. Es zeigt Ihnen die 64 Tasten in einer 8 x 8-Matrix. Diese Anordnung entspricht der elektrischen Verbindung der Tasten. Die Abfrage selbst ist sehr einfach. Der Computer ruft alle senkrechten Spalten einzeln auf, indem er die Zahl, die an der jeweiligen Spalte in Bild 1 steht, in eine spezielle Speicherzelle hineinschiebt. Beim VC 20 ist das 37152, beim Commodore 64 die 56320.

Falls die gedrückte Taste in dieser Spalte liegt, meldet sie sich und die Zahl der waagrechten Zeile wird in die Adresse 37153 (56321) geschrieben. Wenn überhaupt keine Taste gedrückt war, erscheint in 37153 (56321) eine 255.

Theorie und Praxis: Eine untrennbare Einheit

10 IF Sie jetzt fragen, warum die Zahlen so willkürlich und außerdem paarweise gleich sind,
THEN lesen Sie weiter:
GOTO nächsten Absatz.
20 REM ein bißchen Theorie würde nicht schaden: GOTO übernächsten Absatz.

Die Zahlen sind natürlich nicht willkürlich gewählt. Es sind vielmehr die Dezimalwerte von Dualzahlen, die beim Anwählen einer Spalte beziehungsweise durch Drücken einer Taste in den schon genannten Speicherzellen 37152 (56320) und 37153 (56321) entstehen. Bild 2 zeigt den Zusammenhang.

Am Rande der Matrix sehen Sie

jetzt nicht die Zahlen, sondern die beiden Speicherzellen, im folgenden »Register« genannt, und ihre bitweise Verbindung mit Spalten und Zeilen der Tasten. Prinzipiell gilt, daß diejenige Spalte angewählt ist, an deren Stelle eine »Null« im Register 37152 (56320) steht. Das ergibt im Register eine Dualzahl, die entsprechend der Bit-Numerierung zu lesen ist. Eine gedrückte Taste ihrerseits erzeugt eine Null im Register 37153 (56321) an der Stelle, wo ihre Zeile angeschlossen ist. Ich habe in Bild 2 ein Beispiel (Taste mit dem Zeichen »/«) eingezeichnet, und wenn Sie die Dualzahlen kennen, werden Sie beim Vergleich mit den Zahlen in Bild 1 die Übereinstimmung erkennen.

Wie bitte? Sie sagen, daß es aber doch möglich sein müßte, weit mehr als die in Bild 1 gezeigten acht Zahlen im Register 37153 (56321) zu erzeugen, indem man mehr als eine Taste gleichzeitig drückt? Richtig gesehen, das geht in der Tat, und das wollen wir uns auch gleich einmal ansehen. Dazu brauchen wir ein Kochrezept. Ich schlage vor, das Rezept — und auch die noch folgenden — gleich auszuprobieren. Nehmen Sie diesen Artikel und verfolgen Sie die weiteren Zeilen meiner Beschreibung abwechselnd lesend und eintippend.

Die Abfrage der Tastatur wird auf dem Bildschirm dargestellt

Als erstes wollen wir, so wie der Computer es macht, eine Spalte anwählen, zum Beispiel wie in Bild 2, die vierte von rechts. Das ergibt die Dualzahl 11110111, in dezimal die Zahl 247.

```
100   POKE 37152,247
200   PRINT PEEK(37152),
      PEEK (37153)
300   GOTO 100
```

Zeile 200 druckt uns sowohl den von uns gePOKEten (verzeihen Sie mir das schreckliche Computerdeutsch) Wert 247 als auch den in Register 37153 (56321) stehenden Wert aus, der dort erscheint, sobald wir eine Taste in der Spalte 247 drücken. Der Rücksprung in Zeile 300 erzeugt auf dem Bildschirm zwei senkrechte, durchlaufende Zahlenstreifen.

Nach RUN läuft rechts die Zahl 255 (dual = lauter 1er, das heißt keine Taste gedrückt). Drücken Sie nun die »/«-Taste, und es erscheint die 191. Die X-Taste erzeugt 251, die links angeordnete SHIFT-Taste eine 253. Die rechte SHIFT-Taste dagegen zeigt keine Reaktion — ist auch klar, denn sie liegt ja in einer anderen Spalte.

Aber es ist interessant, sich zu merken, daß bei der Tastaturabfrage die beiden SHIFT-Tasten völlig eigenständig behandelt werden, im Gegensatz zu Ihrer Funktion. Sie sehen, ich habe recht gehabt: Alle Tasten sind gleich.

Sie können auch die RUN-STOP-Taste drücken, aber bitte nur ganz, ganz kurz antippen, denn das Programm bleibt natürlich entsprechend ihrer ungeSHIFTeten Funktion sofort stehen. Wenn das Betätigen der Taste aber kurz genug war, dann steht als letzte Zahl die 254 da.

Gleichzeitige Abfrage von zwei Tasten

Jetzt kommt das große Experiment. Nach neuem RUN drücken Sie »Cursor down« und »/« gleichzeitig — es erscheint 63. Der Blick auf Bild 2 zeigt uns für beide Tasten das Bitmuster 00111111 und das ist 63. Es geht also. Probieren Sie ruhig alle Kombinationen aus. Wenn Sie Zeit und die Gelenkigkeit eines Konzertpianisten haben, können Sie alle Zahlen bis 254 erzeugen, auch die Null. Dazu sind alle acht Tasten gleichzeitig zu drücken.

Erwähnenswert ist, daß jetzt das Programm nicht stehen bleibt, obwohl die STOP-Taste gedrückt ist. Aber die SHIFT-Taste ist auch gedrückt, und das ist laut Handbuch die LOAD-RUN-Funktion, die jetzt nicht zum Tragen kommt.

Sie wollen sicher endlich auch einmal die Funktionstasten zur Geltung bringen. Dazu müssen wir aber eine andere Spalte anwählen. Ändern Sie bitte die Zeile 100 ab:
100 POKE 37152, 239

In der Spalte 239 liegt die f-1-Taste. Nach RUN passiert aber was Komisches: Trotz des POKEns von 239 läuft links wieder die 247, genauso wie vorhin. Und außerdem reagieren die Tasten der Spalte 239 nicht, nur die der Spalte 247.

Was passiert da? Es liegt daran, daß nur beim Eintippen eines Programms die Tastaturabfrage genau so funktioniert wie beschrieben. Wenn aber ein Programm abläuft, dann ist der Computer nur an der STOP-Taste interessiert und schiebt deshalb immer wieder eine 247 in das Register 37152 (56320).

Eine Ausnahme gibt es auch hier: INPUT-und GET-Befehle fragen auch die anderen Tasten ab.

Diese erzwungene Vorfahrt der Spalte 247 können wir durch unsere Zeile 100 nicht ändern, denn wir sind ja nach RUN in einem Programmablauf. Die einzige Möglichkeit, andere Spalten POKEn zu können, ergibt sich für uns durch Programmierung in Maschinensprache. Aber darauf will ich jetzt nicht eingehen, sondern erst am Schluß des Aufsatzes.

Wir haben also bei unserem Versuch, das Ergebnis der Befragung schon bei der Stimmenabgabe — sozusagen im Wahllokal — auszukundschaften, Pech gehabt. Das macht aber nichts, denn irgendwann wird ja ein Wahlergebnis offiziell bekanntgegeben. Bei der Tastaturabfrage ist das auch so. Vorher aber wollen wir wenigstens aus dem Teilergebnis, das wir ausspioniert haben, Kapital schlagen und die mehrfache Tastenabfrage der Spalte 247 an einem klaren Beispiel demonstrieren.

Programmsteuerung mit zwei unabhängigen Tasten

Das Programm soll auf dem Bildschirm in den Spalten 6 und 15 zwei senkrechte Bänder mit Sternen darstellen, deren Farbe mit der »X«-Taste und der »/«-Taste unabhängig voneinander, auch gleichzeitig, verändert werden kann.

```
100 PRINT CHR$(147)
```

Diese Zeile löscht den Bildschirm. Diese Methode (ASCII-Code) kennen Sie sicher aus dem Programmier-Handbuch. Ich werde sie aber noch genau behandeln.

```
110 BS=PEEK(648)*256
120 FS=4*(PEEK(36866)
AND 128) + 37888
```

Zeile 110 und 120 machen das Programm unabhängig von Speichererweiterungen. BS und FS sind Variable für die Anfangsadressen des Bildschirm- beziehungsweise des Farbspeichers.

```
130 F=2:G=2
```

Das ist die Anfangsfarbe »rot« für beide Sternreihen.

```
200 FOR Z=0 TO 22
```

Mit Z werden die 23 Zeilen abgezählt.

```
300 POKE BS+5+Z*22,42
310 POKE FS+5+Z*22,F
```

Keine Angst, ich mute Ihnen keine höhere Mathematik zu. Ab Zeile 300 wird der Bildschirm-Code (42) für den Stern (auch das behandle ich später noch) in Spalte 6 (BS+5) für jede Zeile (Z*22) untereinander gePOKEd, dazu die Farbe F in den gleichen Platz des Farbspeichers. Dasselbe gilt in Zeile 400 und 410 für die Spalte 15.

```
400     POKE BS+15+Z*22,42
410     POKE FS+15+Z*22,G
500     AA = PEEK(37153)
510     IF AA=191 OR AA=187
        THEN F=F+1
520     IF AA=251 OR AA=187
        THEN G=G+1
```

Ab Zeile 500 werden die »X«-Taste (191) und die »/«-Taste (251) abgefragt. Wenn eine davon gedrückt ist, wird die Farbzahl F beziehungsweise G um 1 erhöht. Die OR-Funktion, mit welcher der Wert 187 abgefragt wird, erlaubt ein gleichzeitiges Drücken beider Tasten, und es werden sowohl F als auch G erhöht.

Um die Farben zwischen 2 (rot) und 7 (gelb) zu begrenzen, verwenden wir:

```
600 IF F=8 THEN F=2
610 IF G=8 THEN G=2
```

Zum Schluß wird der Zeilenzähler Z weitergesetzt. Wenn er 22 (das heißt die 23. Zeile) erreicht hat, springt das Programm an den Anfang zurück.

```
700 NEXT Z
800 GOTO 200
```

Nun geben Sie RUN ein. Wenn Sie eine der Tasten zu lange, das heißt länger als einen Z-Zyklus, drücken, springt die Farbe weiter. Um das etwas zu erleichtern, können Sie noch eine Verzögerung einbauen:

```
220 FOR T=1 TO 50: NEXT T
```

Ich gebe ja zu, das ist kein gewaltiges Programm. Aber es zeigt Ihnen wenigstens, wie auch in Basic eine mehrfache Tastenabfrage möglich ist.

```
100 PRINT CHR$(147)
110 BS=PEEK(648)*256
120 FS=4*(PEEK(36866)
AND 128) + 37888
130 F=2:G=2
200 FOR Z=0 TO 22
220 FOR T=1 TO 50: NEXT T
300 POKE BS+5+Z*22,42
310 POKE FS+5+Z*22,F
400     POKE BS+15+Z*22,42
410     POKE FS+15+Z*22,G
500     AA = PEEK(37153)
510     IF AA=191 OR AA=187
        THEN F=F+1
520     IF AA=251 OR AA=187
        THEN G=G+1
600 IF F=8 THEN F=2
610 IF G=8 THEN G=2
700 NEXT Z
800 GOTO 200
```

Weiter geht's, und zwar mit dem schon erwähnten Bekanntgeben des Wahlergebnisses. In anderen Worten: Wie wertet der Computer die Tastenabfrage über die Register 37152 (56320) und 37153 (56321) weiter aus?

Sobald der Computer merkt, daß eine Taste gedrückt ist, nimmt er die beiden Zahlen, die in den Registern 37152 (56320) und 37153 (56321) stehen, und wandelt sie in eine Code-Zahl um, die er in der Speicherzelle 203 ablegt. Die Code-Zahl steht auch in der Speicherzelle 197. Mit dem Grund für diese Verdopplung muß ich mich aber erst noch beschäftigen.

Wir bleiben bei Adresse 203. Wie bei der Abfrage der Tastatur-Matrix wollen wir uns den Inhalt dieser Speicherzelle ansehen. Löschen Sie bitte das alte Programm und geben Sie ein:

```
100 PRINT PEEK(203)
200 GOTO 100
```

Nach RUN sehen wir wieder das laufende Zahlenband, jetzt aber mit 64. Das ist die Code-Zahl für »keine Taste gedrückt«. Die »/«-Taste ergibt jetzt 30, die X-Taste 26 und so weiter. Und endlich ist es soweit! Die Funktionstasten reagieren und geben ihre Code-Zahl preis.

Die Funktionstasten reagieren doch!

Probieren Sie alle Tasten durch und schreiben Sie die Code-Zahlen auf die Tasten von Bild 1 oder 2. Jetzt sehen Sie auch die Gesetzmäßigkeit, nach der der Computer die Spalten- und Zeilenzahl der Register ummodelt. Schreiben Sie sich am besten eine komplette Liste der Code-Zahlen für die weitere Verwendung. Die RUN-STOP-Taste läßt sich hier leichter als beim ersten Mal überlisten, natürlich nur mit gleichzeitigem SHIFT.

A propos »gleichzeitig«! Wiederholen Sie das Experiment von vorhin. Hier erleiden wir unseren zweiten Fehlschlag: Mehrfachtasten geben keinen Sinn, denn die Umcodierung verwehrt es uns. Wie gut, daß wir die Methode der Matrix-Abfrage haben, auch wenn sie in voller Eleganz nur in Maschinensprache möglich ist. Doch wie gesagt, davon später.

Zurück zu den einzelnen Tasten. In der Liste der Code-Zahlen fehlen die Tasten RESTORE, SHIFT, C=, CTRL. Sie haben das nicht bemerkt? Dann haben Sie auch noch

nicht die von mir vorgeschlagene Liste gemacht!

Um auch diese Zahlen auf den Bildschirm zu bringen, ergänzen Sie bitte die Zeile 100 in:

```
100   PRINT PEEK (203), PEEK
      (653)
200   GOTO 100
```

Jetzt sehen Sie zwei Zahlenreihen laufen. Zu der bekannten Reihe ist auf der zweiten Hälfte des Bildschirms (bedingt durch das Komma zwischen den PEEKs) eine 0-Reihe gekommen. Drücken Sie jetzt die SHIFT-Taste: Rechts läuft eine 1 — die Code-Zahl dieser Taste.

Die Steuertasten haben ihre Code-Zahl in der Speicherzelle 653

Die C=-Taste erzeugt eine 2, die CTRL-Taste eine 4 (und die natürlich ganz langsam). Drücken Sie mal SHIFT und C= gleichzeitig. Siehe da, bei der Speicherzelle 653 und ihren Steuertasten gibt das einen Sinn. Die Tabelle aller Kombinationen sieht so aus:

keine Taste	0
SHIFT	1
C=	2
SHIFT u. C=	3
CTRL	4
SHIFT u. CTRL	5
CTRL u. C=	6
alle 3 Tasten	7

Während wir das ausprobiert haben, läuft links unbeirrt die 64. Und in der Tat, durch das Aufspalten und Abspeichern der Code-Zahlen in zwei getrennte Speicherzellen können wir beide Tastenarten, nämlich Zeichentasten und Steuertasten, unabhängig voneinander und/oder gleichzeitig abfragen:

Das nutzen wir zum Beispiel bei den Funktionstasten aus. Jede von ihnen hat ihre eigene Code-Zahl in Adresse 203. Aber das gäbe uns nur vier Möglichkeiten, entsprechend der Aufschrift die ungeraden f-Zahlen. Um auch f-2 bis f-8 zu erhalten, kombinieren wir die vier Zahlen in 203 einfach mit SHIFT-Taste gedrückt oder nicht (1 oder 0 in 653).

Aber Sie sehen schon, wie willkürlich das ist, denn wir könnten f-6 auch definieren als Kombination von der dritten Funktionstaste und CTRL (also 55 und 4).

Überhaupt, wir sind gar nicht auf acht Funktionstasten beschränkt, wie es uns durch den Ausdruck eingeredet wird. Die vier Funktionstasten ergeben zusammen mit den acht Kombinationen der Steuertasten 32 mögliche Funktionen. Natürlich gilt das für alle Tasten der Tastatur. Der Computer selbst nutzt allerdings nur wenige Kombinationen aus. SHIFT und C= (3) schaltet alle Buchstaben in Groß-/

Eine »Heimorgel« ganz besonderer Art

Kleinschrift um, die CTRL-Taste mit den Zahlentasten erzeugt die Vordergrund-Farben. Sie haben also viel Raum für phantasievolle Abfragekombinationen. Die Abfrage selbst und ihre Verwendung in einem Programm will ich abschließend mit den Funktionstasten demonstrieren.

Wie man mit dem VC 20 Töne erzeugt, wissen Sie. In Zeile 10 definieren wir das Sopranregister und geben ihm den Namen Z, in Zeile 20 schalten wir die Lautstärke ein. Lautstärke des Fernsehers nicht vergessen!

```
10 Z=36876
20 POKE 36878,10
```

Ab Zeile 40 bis 110 wird jede einzelne Kombination der Code-Zahlen von Funktions- und Steuertasten abgefragt. Sobald sie auftritt, wird ein entsprechender Ton der Tonleiter gePOKEd.

```
30    A= PEEK(203): B=
      PEEK(653)
40    IF A=39 AND B=0 THEN
      POKE Z,131
50    IF A=47 AND B=0 THEN
      POKE Z,145
60    IF A=55 AND B=0 THEN
      POKE Z,157
70    IF A=63 AND B=0 THEN
      POKE Z,162
80    IF A=39 AND B=1 THEN
      POKE Z,172
90    IF A=47 AND B=2 THEN
      POKE Z,181
100   IF A=55 AND B=4 THEN
      POKE Z,189
110   IF A=63 AND B=7 THEN
      POKE Z,193
```

Ein Rücksprung in die Zeile 30 verleiht dem Ton auch die Dauer.

```
120   GOTO 30
```

Damit ein Ton aber nur solange klingt, wie eine Tastenkombination gedrückt ist, schieben wir noch Zeile 35 ein, die das Sopranregister auf 0 (Stille) setzt, sobald »keine Taste« gedrückt ist.

```
35    IF A=64 AND B=0 THEN
      POKE Z,0
```

Jetzt will ich Ihnen natürlich noch die von mir gewählten Tastenkombinationen verraten, damit Sie gezielt »Alle meine Entchen« spielen können. Es gilt der Reihe nach:

Zeile 40	: f-1
Zeile 50	: f-3
Zeile 60	: f-5
Zeile 70	: f-7
Zeile 80	: f-1 u. SHIFT
Zeile 90	: f-3 u. C=
Zeile 100	: f-5 u. CTRL
Zeile 110	: f-7 u. CTRL u. C= u. SHIFT

Ich gebe zu, daß diese Tastenauswahl nicht gerade eine bequeme Klaviatur ergibt.

Ihrem Ehrgeiz ist es überlassen, eine Orgel zu programmieren, die zwar immer noch einstimmig ist, aber eine »normale« Klaviatur hat und auch den vollen Tonumfang des VC 20 ausnutzt. Stellen Sie einfach die Code-Zahlen der Zeichentasten so zusammen, daß eine Tastenreihe die »weißen« Tasten, die darüberliegende Reihe die »schwarzen« Tasten darstellt.

Tastenabfrage und kein Ende: was es sonst noch alles gibt

Mit den Steuertasten können Sie die Oktaven umschalten, mit den Funktionstasten verschiedene Lautstärken. Die Zahlen für die Abfrage entnehmen Sie Ihrer Liste (jetzt wird's aber höchste Zeit, sie zu schreiben).

Statt Töne zu POKEN, können Sie natürlich mit dieser Abfragetechnik der Funktionstasten (und auch der anderen Tasten) alles mögliche per Programm steuern: Raumschiffe abschießen, Textseiten weiterschalten, den Hund rauslassen oder Toast rösten.

Es gibt noch andere Methoden der Tastenabfrage, doch diese will ich Ihnen das nächste Mal erklären.

(Dr. Helmut Hauck)

Literatur

(1) VIC Revealed von Nick Hampshire, Computabits Ltd., 1981

(2) M. Bassman, S. Lederman in COMPUTE!'s first Book of VIC. Compute! Books Publication, 1982

(3) A. Grant in VIC COMPUTING, Dezember 1982

(4) A. Dripke, VC 20 Spiele-Buch 1, Computer Life Verlag, 1983

Reise durch das Wunderland der Grafik

Begleiten Sie uns auf einer abenteuerlichen, aber ungefährlichen Reise durch das Wunderland der Grafik. Entdecken Sie die (fast) grenzenlosen Grafikmöglichkeiten des Commodore 64.

Ist es Ihnen auch so ergangen: Sie lesen Testberichte über den Commodore 64, seine hervorragenden Grafikmöglichkeiten, eine Auflösung von 320 x 200 Punkten, und Ihnen schweben die phantastischen Abbildungen von Computergrafiken vor, die Sie nun auch alle selbst realisieren können, wenn Sie diesen Computer in Händen halten. Dann, nach mehr oder weniger vielen Anstrengungen, sitzen Sie vor Ihrem eigenen Commodore 64, neben sich das 170 Seiten dicke Handbuch, und arbeiten sich durch alles hindurch. Aber wo ist diese schöne Grafik? Nachdem Sie frustriert ein paar Sprites über den Bildschirm ziehen ließen und die Ballspiele aus dem Handbuch anfangen, Sie zu langweilen, geht die Suche los, wie man denn nun eine hübsche dreidimensionale Grafik auf den Bildschirm zaubern kann: Im Handbuch ist nichts zu finden. Dann gibt es nur zwei Möglichkeiten: Entweder Sie lassen die Grafik Grafik sein oder Sie beginnen eine Odyssee durch Buchläden, Computerzeitschriften und auch durch die Speicherzellen Ihres Computers, um sie nach langen Irrwegen endlich zu finden: die hochauflösende Grafik.

Wenn Ihre Barschaft es erlaubt, können Sie sich natürlich einiges an Schweiß ersparen: Inzwischen wird ja eine Reihe von mehr oder weniger brauchbarer Grafik-Software angeboten. Aber was Sie nicht bezahlen können, ist eine Menge von Erkenntnissen über die Möglichkeiten, die — verborgen hinter dornigen POKE-Hecken — darauf warten, von Ihnen wie Dornröschen wachgeküßt zu werden. Glauben Sie mir, diese Küsse sind es wert, sich in das Byte-Gewirr zu stürzen, zumal ich versuchen werde, Ihnen dazu den von mir schon gebahnten Weg hier und in den kommenden Folgen zu zeigen. Bei der Gelegenheit werden Sie feststellen, daß Sie nicht nur Dornröschen (die hochauflösende Grafik) wachgeküßt haben, sondern — erinnern Sie sich an die Gebrüder Grimm — auch das ganze Volk im Schloß fing an zu leben. Mit nüchternen Worten: Sie machen sich dabei eine Menge anderer, sonst schlafender Eigenschaften Ihres Computers zunutze.

65535
64 KByte RAM
0

Bild 2. Die 64 KByte RAM pur

VIC-II-Chip Grafikmöglichkeiten
Darstellung von Zeichen
normaler Zeichen-Modus
Mehrfarben Zeichen-Modus
Modus mit verändertem Hintergrund
eingebauter Zeichensatz
programmierbarer Zeichensatz
Sprites
normale Sprites
mehrfarbige Sprites
Hochauflösungsmodus
normale Hochauflösung
mehrfarbige Hochauflösung

Bild 1. Die Vielfalt der Grafikmöglichkeiten des Commodore 64

Die Grafikmöglichkeiten des C 64

Noch einige Worte, bevor wir an die Arbeit gehen: Wissen Sie eigentlich, welche Grafik-Vielfalt der Commodore 64 hat? In Bild 1 ist sie aufgeführt.

Im Handbuch finden Sie davon nur zwei angegeben: Den »normalen« Zeichensatz und die »normalen« Sprites. Zu dem Schema in Bild 1 gehören eigentlich noch einige Kleinigkeiten, auf die wir noch stoßen werden. So kann beispielsweise der Bildschirm auf verschiedene Grafikarten aufgeteilt werden und so weiter. Aber um so weit zu gelangen, müssen wir uns erst eine Weile durch die Byte-Dornen gehauen haben.

Sie dürfen schon Ihren Computer anschalten, denn wir werden bei der nun folgenden Reise durch das

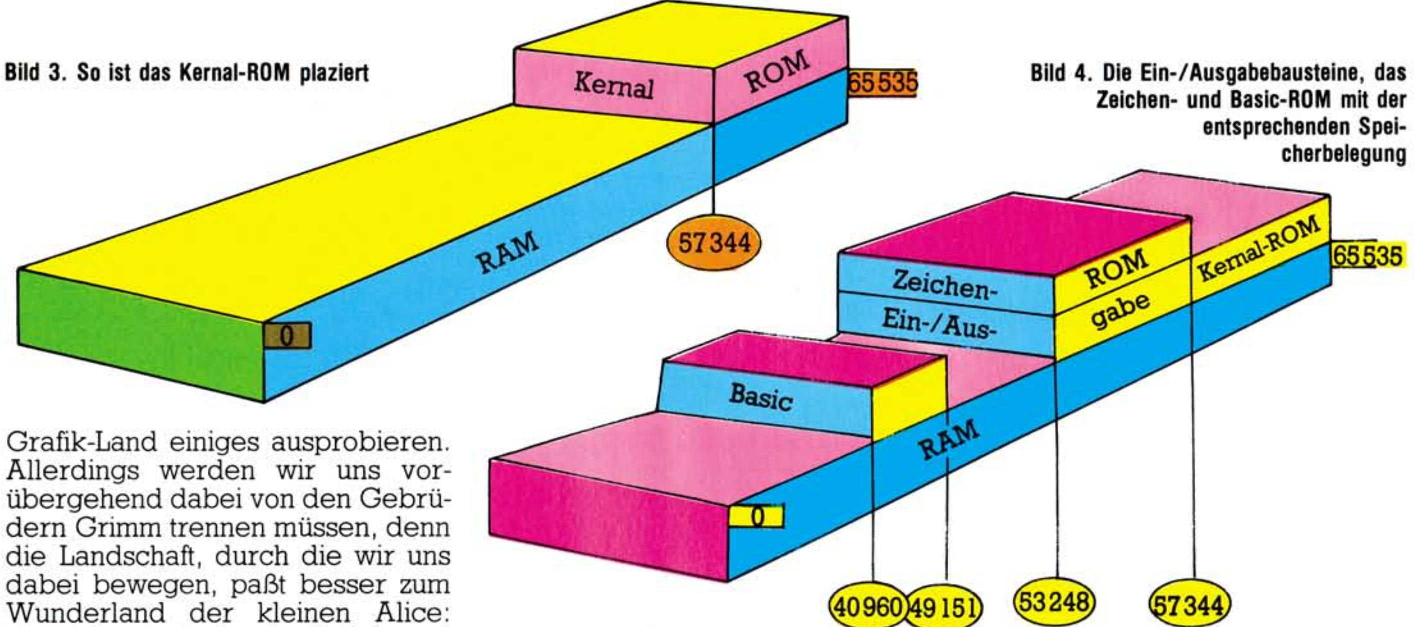


Bild 3. So ist das Kernal-ROM plaziert

Bild 4. Die Ein-/Ausgabebausteine, das Zeichen- und Basic-ROM mit der entsprechenden Speicherbelegung

Grafik-Land einiges ausprobieren. Allerdings werden wir uns vorübergehend dabei von den Gebrüdern Grimm trennen müssen, denn die Landschaft, durch die wir uns dabei bewegen, paßt besser zum Wunderland der kleinen Alice: Ganze Landschaftsteile sind da und doch nicht da, Gebäude verschwinden und andere tauchen wieder auf, die Zeit wird gedehnt, Spiegelbilder erscheinen.

Wir fangen zunächst mal damit an, den Ast, auf dem wir sitzen, abzusägen. Damit Sie sich trotzdem keinen Schaden zufügen, sollten Sie vorher noch alle Programme, die Sie eventuell noch im Computer haben, auf Kassette oder Diskette abspeichern. Erledigt? Dann geben Sie doch jetzt mal folgendes ein:

POKE 1,PEEK(1) AND 252
»RETURN«

Jetzt ist Ihr Computer scheintot. Kein Cursor mehr, keine Reaktion auf Tastendrucke. Aber dafür haben Sie jetzt tatsächlich die 64 KBytes RAM (Worterklärungen siehe Kasten), die in der Kaltstartmeldung des C 64 angekündigt sind, zur freien Verfügung (Bild 2). Nur ist nichts damit anzufangen! Die 65536 freien Bytes Speicherkapazität liegen wie jungfräulicher Ackerboden vor uns und wir Benutzer sind ihnen völlig egal: Es muß also außer dem, was über einen Adreßbus von 16 Bit normalerweise erreichbar ist, noch etwas anderes vorhanden sein, etwas, das uns die Kommunikation mit unserem Computer erlaubt.

Natürlich ist das auch jetzt vorhanden, nur der 64 sieht es nicht. Das zwingt uns leider dazu, einige für ihn verschwundene Gebäude durch Aus- und Einschalten schlagartig wieder sichtbar zu machen. Welche Gebäude sieht der Computer jetzt wieder?

Da ist zunächst einmal das Betriebssystem (auch Kernal-ROM genannt). Alle Hausnummern unserer Byte-Straße von 57344 bis 65535 haben noch eine Etage außer dem RAM-Erdgeschoß: Im ersten Stock liegt dort das Kernal-ROM (Bild 3).

Dieses Kernal-ROM ist sozusagen der Organisator unseres Computers — nichts geht ohne ihn, wie wir ja eben, als er weggeschaltet war, gesehen haben. Allerdings braucht auch der beste Organisator noch einige andere lebenswichtige Partner. Damit wir überhaupt mit dem Computer in Verbindung treten können, sind noch einige weitere Hausnummern zumindest einstöckig (Bild 4).

53248 bis 57343. Ein- und Ausgabebausteine
40960 bis 49151. Basic-ROM

Es gibt sogar Häuser mit einem zweiten Stock.

53248 bis 57343. Zeichen-ROM

Zum Zeichen-ROM werden wir später kommen und die Ein- und Ausgabebausteine werden uns eine ganze Weile beschäftigen. Ohne Basic-ROM könnten wir nur in Maschinensprache unseren C 64 programmieren und eben nicht in Basic.

Wie kann unser Computer diese anderen Etagen nützen? Es sind ja insgesamt statt 64 KByte jetzt 88 KByte oder exakt 90112 Zimmerfluchten zu je 8 Bit, auf die man ge-

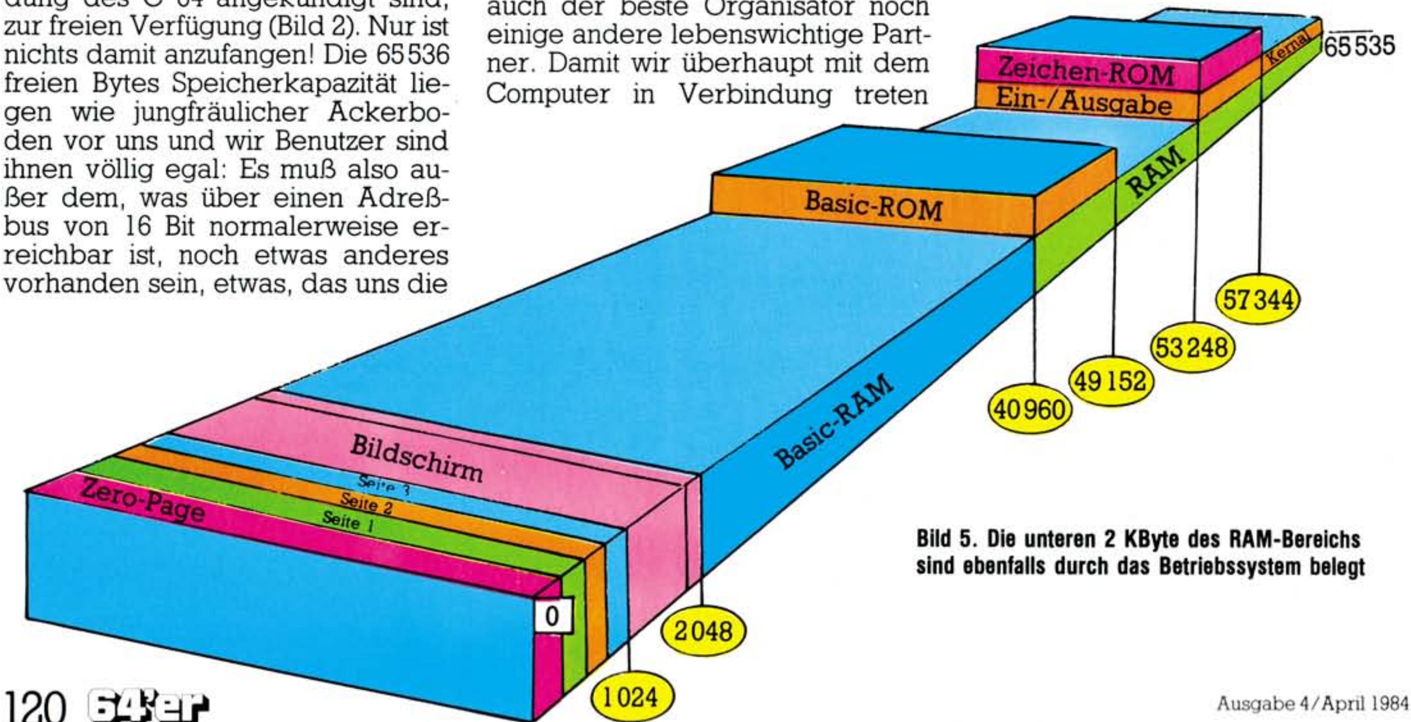
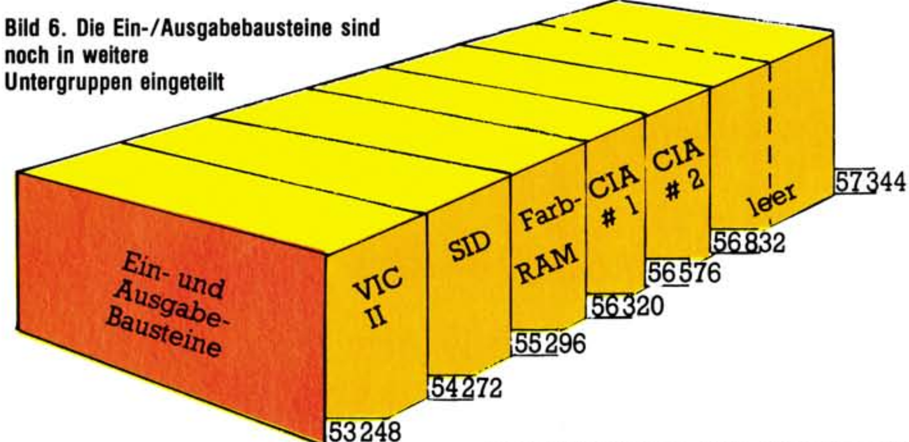


Bild 5. Die unteren 2 KByte des RAM-Bereichs sind ebenfalls durch das Betriebssystem belegt

langen können muß. Man kann sich das so vorstellen, daß zum Beispiel zwischen den Hausnummern 53248 und 57343 einen Augenblick lang die Ein- und Ausgabebausteine stehen, dann verschwinden sie und im nächsten Augenblick steht das Zeichen-ROM dort, dann wieder die Ein-/Ausgabebausteine und so weiter. Also tatsächlich ein Wunderland, das wir an dieser Byte-Straße finden. Gesteuert wird dieses Auftauchen und Verschwinden vom Betriebssystem. In Wirklichkeit bleibt alles an seinem Platz.

Bild 6. Die Ein-/Ausgabebausteine sind noch in weitere Untergruppen eingeteilt



Man sollte meinen, daß der Commodore 64 durch diese ganzen Zaubereien, denen er sich da widmen muß, wenig Zeit für uns Benutzer hat! Aber weit gefehlt, unser Computer ist so schnell, daß für uns seine Zeit gedehnt aussieht. Der Puls des Computers rast mit zirka 1 Million Schlägen pro Sekunde, während unser Puls rund einhalbmal in der Sekunde schlägt: In der Zeit also, in der unser Augenlid einmal zwinkert, hat der Computer schon tausende von Operationen vorgenommen und steht gewissermaßen mit den Fingern trommelnd bereit, unser Kommando endlich zu empfangen. Eigentlich langweilt er sich die meiste Zeit. Wie man seine Leistungsfähigkeit effektiver als mit Basic-Programmen ausnutzen kann, dazu werden wir in dieser Serie auch noch kommen.

Zunächst wollen wir uns mal ein wenig umsehen in unserem Speicher. Dazu kann das angefügte Programm »SpeiLu« benutzt werden (siehe Listing). Ziemlich primitiv, für unsere Zwecke zunächst aber schon ausreichend, ist dieses kleine Programm:

```
10 INPUT"STARTADRESSE";A
20 FOR I=A TO A+255:PRINT
   PEEK(I);:NEXT
30 GET A$:IF A$="" THEN 30
40 IF A$="←" THEN A=I:GOTO
   20
```

Geben Sie nach dem RUN jetzt mal als Startadresse 6000 ein. Es erscheinen Blöcke von Nullen und Blöcke von 255ern (meistens). Wenn Sie "←" drücken, kommen die nächsten 256 Bytes auf den Bildschirm und so weiter. So sieht ein leerer Speicher aus. Drücken Sie irgendeine Taste (außer "←") und starten Sie mit RUN erneut. Mit der Eingabe von 2048 blicken wir in die ersten 256 Bytes unseres Basic-RAMs. Der wüste Zahlensalat in der oberen Hälfte des Bildschirms ist die Computer-Version unseres Programms. Danach ist dann wieder leerer Speicher zu sehen.

Im Speicher ist einiges los

Wieso eigentlich 2048 als Start des Basic-RAMs? Warum nicht 0? Sehen wir uns doch mal mit ein bißchen Geduld den RAM-Bereich von 0 bis 2048 an, also Starten des Programms und Eingeben von 0: Wir sehen einen nahezu vollen Speicher. Das ist die sogenannte Zero-Page, zu deutsch Null-Seite. Voll ist die Seite, weil sie uns das Betriebssystem abgezwackt hat, um dort eine Reihe wichtiger Werte zu speichern. Wie wichtig das ist, haben wir gesehen, als wir den Wert 55 des ersten Byte (auf dem Bildschirm jetzt die zweite Zahl oben links) durch unser Astabsägen verändert haben. Wenn wir jetzt "←" drücken, sehen wir die nächste Seite (page 1) auf dem Bildschirm. Auch diese Seite — obwohl sie jetzt größtenteils leer ist (Nullen und 255er-Blöcke) — gehört dem Betriebssystem: Es ist der sogenannte Prozessorstapelspeicher. Drücken wir nochmal "←", dann erscheint Seite 2 (Adresse 512 bis 767). Hier ist zwar auch vieles leer (viele Nullen), aber wenn wir uns recht erinnern, sah der normale leere Speicher anders aus. Auch

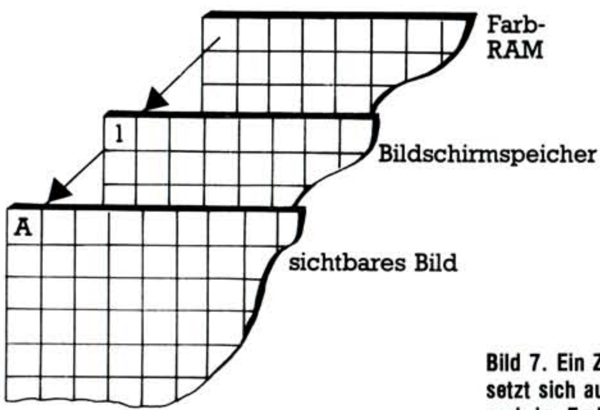


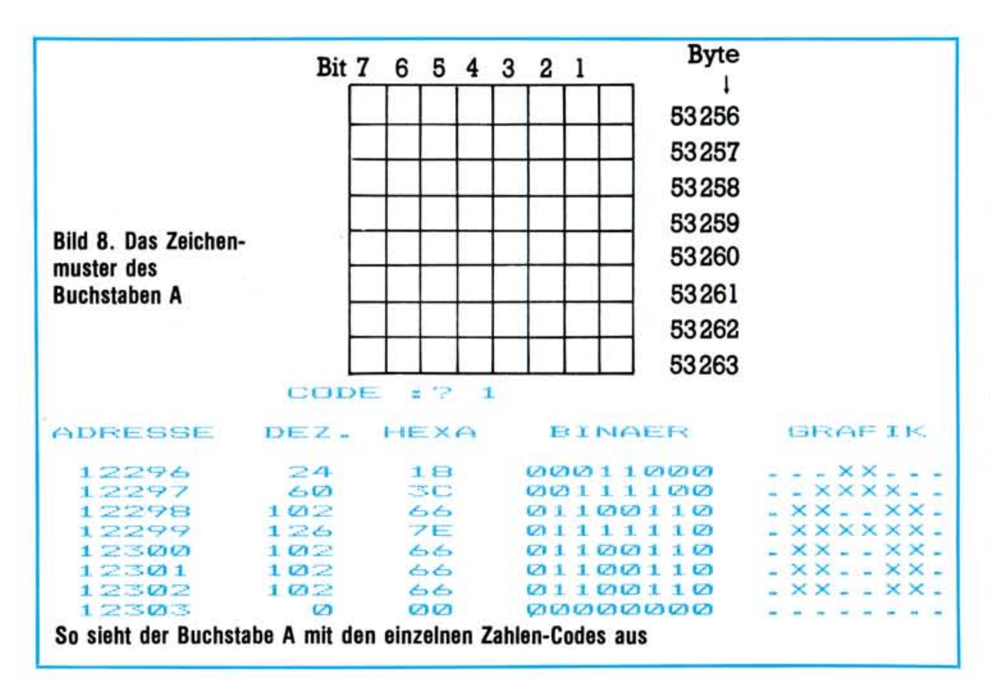
Bild 7. Ein Zeichen auf dem Bildschirm setzt sich aus der Bildschirm- und der Farbinformation zusammen.

diese Seite hat der Computer unserem Zugriff durch das normale Basic entzogen und speichert dort einige wichtige Angaben. Die Seite 3 erfüllt einen ähnlichen Zweck und außerdem befindet sich dort von 828 bis 1019 noch der Kassettenpuffer. Damit hat uns das Betriebssystem unseres Computer also schon das erste KByte des Speichers gemopst. Wenn Sie sich entsinnen, habe ich vorhin erwähnt, daß das Basic-RAM bei 2048 beginnt. Wie sieht es also im Bereich des zweiten KByte aus? Wenn wir uns mittels "←" die nächsten vier Seiten ansehen, marschieren stramme Kolonnen von Zahlen zwischen 48 und 57 (und viele 32) auf. Das sind Bildschirm-Codes von Zahlen und Leerstellen: Hier haben wir den Bildschirmspeicher mit insgesamt 1000 Bytes und dazu noch einige Bytes, die uns bei den Sprites beschäftigen werden. Jetzt sind wir wieder beim Basic-RAM ab 2048 angelangt. Haben Sie noch Lust? Dann probieren Sie noch ein bißchen weiter und sehen sich zum Beispiel das Basic-ROM zwischen 40960 und 49151 an oder das Betriebssystem oder ...

Dabei werden Sie dann nochmal einen freien RAM-Bereich zwischen 49152 und 53247 finden (Bild 5), der aber normalerweise nicht für Basic erreichbar ist. Jetzt ken-

Tabelle 1. Registerübersicht des VIC-II-Chips

Register	Adresse	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	53248	X-Position des Sprite Nr. 0. Dazu muß Register 16 beachtet werden							
1	53249	Y-Position des Sprite Nr. 0							
2	53250	X-Position des Sprite Nr. 1. Auch dazu, wie zu allen folgenden Sprites, muß Register 16 beachtet werden.							
3	53251	Y-Position des Sprite Nr. 1							
4	53252	X-Position des Sprite Nr. 2. s. o.							
5	53253	Y-Position des Sprite Nr. 2							
6	53254	X-Position des Sprite Nr. 3. s. o.							
7	53255	Y-Position des Sprite Nr. 3							
8	53256	X-Position des Sprite Nr. 4. s. o.							
9	53257	Y-Position des Sprite Nr. 4							
10	53258	X-Position des Sprite Nr. 5. s. o.							
11	53259	Y-Position des Sprite Nr. 5							
12	53260	X-Position des Sprite Nr. 6. s. o.							
13	53261	Y-Position des Sprite Nr. 6							
14	53262	X-Position des Sprite Nr. 7. s. o.							
15	53263	Y-Position des Sprite Nr. 7							
16	53264	Spr. 7, msb X-Pos.	Spr. 6, msb X-Pos.	Spr. 5, msb X-Pos.	Spr. 4, msb X-Pos.	Spr. 3, msb X-Pos.	Spr. 2, msb X-Pos.	Spr. 1, msb X-Pos.	Spr. 0, msb X-Pos.
17	53265	msb des Raster-registers (Reg. 18)	Schaltbit für veränderten Hintergrund-farbmodus 1 = einge-schaltet	Schaltbit für Hochauflö-sungsmodus 1 = einge-schaltet	Schaltbit für Bildschirm »aus« 0 = normaler Bildschirm 1 = Bild-schirmfarbe gleich Hinter-grundfarbe	Schaltbit für Zeilenzahl 0 = 24 Zeilen 1 = 25 Zeilen	Wert der Zeilenverschiebung in Y-Richtung beim Smooth Scrolling		
18	53266	Rasterregister. Dazu kommt das msb in Bit 7, Register 17							
19	53267	Lichtgriffel X-Position							
20	53268	Lichtgriffel Y-Position							
21	53269	Ein- und Ausschalten von Sprites. 0 = Sprite aus. 1 = Sprite an Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0
22	53270			Reset-Bit, muß 0 sein, damit VIC-II-Chip arbeitet	Schaltbit für Mehrfarb-modus 1 = einge-schaltet	Schaltbit für Spaltenzahl 0=38 Spalten 1=40 Spalten	Wert der Spaltenverschiebung in X-Richtung beim Smooth Scrolling		



ADRESSE	DEZ.	HEXA	BINAER	GRAFIK
12296	24	18	00011000	---XX---
12297	60	3C	00111100	--XXXX--
12298	102	66	01100110	-XX--XX-
12299	126	7E	01111110	-XXXXXX-
12300	102	66	01100110	-XX--XX-
12301	102	66	01100110	-XX--XX-
12302	102	66	01100110	-XX--XX-
12303	0	00	00000000	--------

Bild 8. Das Zeichenmuster des Buchstaben A

So sieht der Buchstabe A mit den einzelnen Zahlen-Codes aus

nen wir — bis auf einige weitere Merkwürdigkeiten, zum Beispiel die versprochenen Spiegelbilder — unseren Computerspeicher schon ganz gut und können uns dem für die Grafik wichtigsten Speicherteil zuwenden: den Ein- und Ausgabebausteinen.

Der VIC-II-Chip

Verschaffen wir uns zunächst einen Überblick:

☐ Der Video-Interface-Controller 6567 (VIC-II) liegt zwischen den Hausnummern 53248 und 54271. Genaugenommen ist die letzte Register-Hausnummer allerdings schon 53294. Alle Register liegen tatsächlich nur auf 47 von den 1024 Hausnummern.

☐ An den VIC-II-Chip schließt sich

Register	Adresse	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
23	53271	Sprite-Vergrößerung in Y-Richtung. 0 = normale Größe. 1 = doppelte Größe.							
		Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0
24	53272	Startadresse des Bildschirmspeichers				— Startadresse des Speicherbereichs, in dem die Zeichen als Punktmatrizen abzurufen sind. — Startadresse der Bit-Map			
25	53273	Interrupt-Flaggen-Register Interrupt				Lichtgriffel-Interrupt-Flagge	Sprite/Sprite-Kollision	Sprite/Hintergrund-Kollision	Raster-Interrupt-Flagge
26	53274	Interrupt-Masken-Register Interrupt				Lichtgriffel-Interrupt-Maske	Sprite/Sprite-Koll.-Maske	Sprite/Hintergrund-Kollision Maske	Raster-Interrupt-Maske
27	53275	Sprite/Hintergrund-Prioritätenregister. 0 = Sprite hat Priorität. 1 = Hintergrund hat Priorität							
		Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0
28	53276	Sprite-Mehrfarbmodus-Register. 0 = Normaldarstellung. 1 = Mehrfarbmodus-Darstellung							
		Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0
29	53277	Sprite-Vergrößerung in X-Richtung. 0 = normale Größe. 1 = doppelte Größe							
		Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0
30	53278	Sprite/Sprite-Kollision. 0 = keine Berührung. 1 = Berührung							
		Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0
31	53279	Sprite/Hintergrund-Kollision. 0 = keine Berührung. 1 = Berührung							
		Sprite 7	Sprite 6	Sprite 5	Sprite 4	Sprite 3	Sprite 2	Sprite 1	Sprite 0
32	53280	unbenutzt				Farbe des Bildschirmrahmens			
33	53281	unbenutzt				Hintergrundfarbe Nr. 0 (normale Hintergrundfarbe)			
34	53282	unbenutzt				Hintergrundfarbe Nr. 1			
35	53283	unbenutzt				Hintergrundfarbe Nr. 2			
36	53284	unbenutzt				Hintergrundfarbe Nr. 3			
37	53285	unbenutzt				Sprite-Mehrfarben-Register Nr. 0			
38	53286	unbenutzt				Sprite-Mehrfarben-Register Nr. 1			
39	53287	unbenutzt				Sprite 0, Farbe			
40	53288	unbenutzt				Sprite 1, Farbe			
41	53289	unbenutzt				Sprite 2, Farbe			
42	53290	unbenutzt				Sprite 3, Farbe			
43	53291	unbenutzt				Sprite 4, Farbe			
44	53292	unbenutzt				Sprite 5, Farbe			
45	53293	unbenutzt				Sprite 6, Farbe			
46	53294	unbenutzt				Sprite 7, Farbe			

das Sound Interface Device 6581 (SID) an, welches von Hausnummer 54272 bis 55295 reicht. Das ist ein ebenfalls sehr verlockendes Nachbaranwesen (Musikliebhaber kommen hier auf ihre Kosten), welches wir bei dieser Gelegenheit aber nicht besuchen wollen.

□ Von 55296 bis 56319 (genauer eigentlich nur bis 56295) liegt das Farb-RAM, Dornröschens hauseigene Malerei, die wir noch bemühen werden.

□ Stippvisiten werden wir uns erlauben beim Pförtner des Schlosses, der seine Wache bei den Hausnummern 56320 bis 56575 stehen hat, dem sogenannten Complex Interface Adapter 6526 (CIA Nr. 1). Die 1 rührt daher, daß er noch einen Kollegen hat, der das Revier von Adresse 56576 bis 56831 bewohnt und logischerweise CIA Nr. 2 heißt.

□ Sozusagen Baugrund für Erweiterungen findet man noch zwischen den Speicheradressen 56832 und 57343. Die einzelnen Abteilungen sind in Bild 6 aufgeschlüsselt.

Von nun an wird uns Dornröschens Schloß, der VIC-II-Chip, ständig beschäftigen. Damit die Orientierung leichter fällt, ist die Tabelle 1 abgebildet, in der alle Registerinhalte wie auf einem Grundriß verzeichnet sind. Auf den ersten Blick sieht das zugegebenermaßen reichlich verwirrend aus — lassen Sie sich nicht erschrecken.

Sie stehen jetzt sozusagen mitten im Dornengestrüpp, und wenn wir gemeinsam den Weg hindurchgefunden haben, wird Ihnen alles verständlich sein, was da steht. Fangen wir mit der Hausnummer 53272 an:

Wenn sie Lust haben, können Sie sich den Inhalt der Adresse 53272 einmal mit dem beigefügten Programm »SpeiLu« betrachten. Aus der Registerübersicht werden Sie beim Nachschlagen sehen, daß die Bit (Zimmer) 7-4 den Ort des Bildschirms im Speicher anzeigen und die Bit 3-0 im Normalfall etwas damit zu tun haben, wo die Zeichen (Buchstaben, Zahlen, Grafikzeichen etc.) abrufbar sind. Bevor wir daran gehen, dieses Byte zu verändern, wollen wir es im Urzustand erstmal unter die Lupe nehmen. Wenn Sie
PRINT PEEK(53272)
eingeben, werden Sie den Wert 21 ausgedruckt bekommen. Haben Sie sich diesen Speicherplatz mittels »SpeiLu« angesehen, dann fanden Sie etwas wie
00010101
was der Binärausdruck der Dezimalzahl 21 ist. Aber dazu kommen wir noch. Das Betriebssystem setzt nach dem Einschalten das Byte 53272 automatisch auf diesen Wert, und wenn Sie sich an die Speicherreise mit dem kleinen Vierzeilen-Programm erinnern, dann wissen Sie auch noch, daß der Bildschirmspeicher damit auf die Startadresse 1024 festgelegt ist.

Wie Sie ebenfalls wissen, hat der Commodore 64 im Normalfall 40 Zeichen pro Zeile und 25 Zeilen, was insgesamt 1000 Zeichen pro Bildschirm ergibt. Deswegen hat der Bildschirmspeicher eine Ausdehnung von 1000 Bytes: von 1024 bis 2023. Die Aufteilung dieser 1000 Speicherplätze auf den Bildschirm ersehen Sie aus dem Handbuch auf Seite 138. Was ist nun drin in den Bildschirmspeicherstellen? Probieren wir es aus! Tippen Sie doch mal ein:
»shift + clear home« ABC »Return«

Natürlich taucht jetzt eine Fehlermeldung auf, die uns aber nicht kümmern soll. Jetzt steht ganz links oben (in Speicherplatz 1024) das A, dann B (1025) und C (1026). Nun wollen wir mal sehen, was der Computer sich merkt:
PRINT PEEK(1024), PEEK(1025, PEEK(1026) »Return«
Es erscheint 1 2 3 .

Wenn also auf dem Bildschirm ein A vorhanden ist, hat der Computer in der dazugehörigen Stelle seines Speichers eine 1 stehen, bei B eine 2, bei C eine 3 und so weiter. Lassen Sie uns den Bildschirm nochmal löschen mit »shift + clear home«. Dann gehen wir mit dem Cursor etwas abwärts und poken diese Kennzahlen in den Bildschirmspeicher:
POKE 1024,1:POKE 1025,2:
POKE1026,3 »Return«

Nach dem Return ist anscheinend nichts passiert. Erst wenn Sie mit dem Cursor dorthin fahren, wo eigentlich ABC stehen sollte, tauchen diese Buchstaben unter dem Cursor auf. Der Grund für dieses Verhalten liegt sicherlich darin, daß die Kennzahl 1 in der Speicherzelle 1024 alleine nicht genügt, das A sichtbar zu machen. Es hat automatisch die Farbe des Hintergrundes. Einem Zwilling des Bildschirmspeichers sind wir bei den Ein- und Ausgabebausteinen schon begegnet: dem Farb-RAM zwischen

Block	Adressenbereich	Zeichen	Muster abrufbar im Programm mit Code
0	53248 — 53759	Satz 1 von 0 bis 63 (0 bis ?)	0 — 63
	53760 — 54271	Satz 1 von 64 bis 127 (Grafikz.)	64 — 127
	54272 — 54783	Satz 1 von 0 bis 63 (Reversed)	128 — 191
	54784 — 55295	Satz 1 von 64 bis 127 (Reversed)	192 — 255
1	55296 — 55807	Satz 2 von 0 bis 63 (kleine Buchst.)	256 — 319
	55808 — 56319	Satz 2 von 64 bis 127 (Großbuchst. + Grafikz.)	320 — 383
	56320 — 56831	Satz 2 von 0 bis 63 (Reversed)	384 — 447
	56832 — 57343	Satz 2 von 64 bis 127 (Reversed)	448 — 511

Tabelle 2. Inhalt des Zeichen-ROMs

Damit Sie nicht über Begriffe stolpern, sind sie hier erklärt:

RAM	= Random Access Memory = Speicher für beliebigen Zugriff, also Schreiben und Lesen (POKE und PEEK) möglich.
ROM	= Read Only Memory = Speicher ist nur zum Lesen (PEEK)
Speicher	kann man sich vorstellen als lange Straße mit meist ebenerdigen Häusern und Hausnummern von 0 bis 65535
Byte	Ein Haus dieser Straße mit acht Zimmern. Man numeriert sie durch von 0 bis 7.
Bit	Ein Zimmer eines solchen Hauses. Es ist entweder etwas drin (Bit gesetzt, also = 1) oder nichts drin (Bit gelöscht, also = 0)
Adreßbus	Eine Art Kabinentaxi, das alle 65536 Häuser durch Angabe der Hausnummer ansteuern kann. Eine höhere Zahl als 65535 kann nicht angegeben werden.
1 KByte	= Einmal 1024 Bytes
1 page	= Eine Seite = ein Viertel von 1 KByte = 256 Bytes

```
1 REM*************************************************************************
2 REM*                                                                       *
3 REM*                                                                       *
4 REM*                                S P E I L U                            *
5 REM*                                                                       *
6 REM*                                                                       *
7 REM*   MANFRED THOMA  +  HEIMO PONNATH  2102 HAMBURG 93  VERINGSTRASSE 82  *
8 REM*                                                                       *
9 REM*************************************************************************
10 POKE52,48:POKE56,48:POKE53280,6:POKE53281,6:POKE646,1
20 TE$=" ADRESSE  DEZ. HEXA   BINAER    GRAFIK"
100 PRINTCHR$(147)"  DARSTELLUNG VON ZEICHEN UND SPEICHER"
105 PRINT:PRINTCHR$(18);TAB(10);"====================="
110 PRINTCHR$(18):TAB(10)"THOMA/PONNATH HAMBURG"
115 PRINTCHR$(18);TAB(10);"====================="
120 PRINT:PRINT:PRINT"     (1) EINSEHEN IN EINEN SPEICHER"
130 PRINT:PRINT"     (2) DARSTELLUNG EINES ZEICHENS"
150 PRINT:PRINT:PRINT"        BITTE KENNZIFFER WAEHLEN !"
160 GETA$:IFA$=""THEN160
170 A=VAL(A$):IFA<0ORA>2THEN160
180 ONAGOSUB1000,2000
190 GOTO100
1000 PRINTCHR$(147);CHR$(18);TAB(2);"DARSTELLUNG EINES SPEICHERPLATZES:"
1010 PRINT:PRINT" SPEICHERADRESSE (0-65535) EINGEBEN"
1020 PRINT"ZURUECK MIT ZAHL AUSSERHALB 0 UND 65535":PRINT
1030 INPUT"ADRESSE :";AD:IFAD<0ORAD>65535THENRETURN
1040 DE=PEEK(AD):PRINT:PRINT:GOSUB10000:GOSUB20000:PRINTTE$:PRINT:GOSUB30000
1050 PRINT:PRINT:GOTO1010
2000 PRINTCHR$(147):CHR$(18):TAB(3);"ZEICHEN-DARSTELLUNG (CHARACTER)"
2010 IFTS=0THENGOSUB40000
2020 PRINT:PRINT"      GEBE DEN 'BILDSCHIRM CODE' DES"
2030 PRINT"      DARZUSTELLENDEN ZEICHENS EIN"
2040 PRINT"   = SIEHE HANDBUCH SEITE 133 - 134 ="
2050 PRINT:PRINT"  ZURUECK MIT ZAHL AUSSERHALB 0 UND 511":PRINT
2060 PRINT,:INPUT"CODE :";A:IFA<0ORA>511THENRETURN
2070 PRINT:PRINTTE$:PRINT
2080 FORAD=12288+8*ATO12288+8*A+7
2100 DE=PEEK(AD):GOSUB10000:GOSUB20000:GOSUB30000:NEXTAD
2110 GETA$:IFA$=""THEN2110
2120 RETURN
10000 HE$="":H$="0123456789ABCDEF":D=INT(DE/16):HE$=MID$(H$,D+1,1):D=DE-D*16
10010 HE$=HE$+MID$(H$,D+1,1):RETURN
20000 BI$="":DI=DE
20010 DI=DI/2:D$="0":IFDI<>INT(DI)THEND$="1"
20020 DI=INT(DI):BI$=D$+BI$:IFDI>0THEN20010
20030 IFLEN(BI$)<8THENBI$="0"+BI$:GOTO20030
20040 RETURN
30000 PRINTTAB(7-LEN(STR$(AD)))AD;TAB(13-LEN(STR$(DE)))DE;TAB(16)HE$;TAB(21)BI$;
30010 FORI=1TO8:W$=MID$(BI$,I,1)
30020 IFW$="1"THENPRINTTAB(30+I)CHR$(18);" ";CHR$(146);:GOTO30040
30030 PRINTTAB(30+I)".";
30040 NEXTI
30050 PRINT: RETURN
40000 PRINT:PRINT"KOPIEREN DER ZEICHEN INS RAM (AB 12288)"
40010 PRINT"              BITTE WARTEN"
40020 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251
40030 FORI=0TO4095:POKE12288+I,PEEK(53248+I):NEXTI
40040 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1
40050 POKE53272,(PEEK(53272)AND240)+12:TS=1:RETURN

READY.
```

Listing. Das Programm »SpeiLu« (Speicherlupe)

55296 und 56295. Wie er aufgeteilt ist (Bild 7) steht im Handbuch Seite 139 zusammen mit den Farbkennzahlen.

Wenn wir jetzt zum Beispiel noch eingeben:

POKE 55296,1:POKE 55297,3:POKE 55298,7 »Return«

dann sehen wir ein weißes A, ein cyanfarbenes B und ein gelbes C.

Übrigens, wenn Ihnen die aktuelle Farbe des Cursors oder der gerade verwendeten Zeichen nicht gefällt, dann probieren Sie doch mal

POKE 646, Farbkennzahl.

Und weil wir gerade bei den Farben sind, die Speicherzellen 53280 und 53281 steuern, mit Farbkennzahlen belegt, die Rahmen- und die Hintergrundfarbe. Mir persönlich gefällt zum Beispiel folgende Kombination sehr gut (auf Schwarzweiß-Bildschirm)

POKE 53280,11:POKE 53281,11: POKE 646,0 »Return«

Nun zu den Zeichen. Woher weiß der Computer, daß er ein A drucken muß, wenn eine 1 im Bildschirmspeicher steht? Das sagt ihm das Betriebsystem. Es teilt ihm mit, daß im Byte 53272 des VIC-II-Chips eine Kennzahl steht (Bit 1-3, Bit 0 wird nicht beachtet), die ihm wiederum sagt, wo die Muster für alle Zeichen zu finden sind. Merkwürdigerweise deutet diese Kennzahl auf eine Startadresse der Zeichenmuster von 4096! Das Zeichen-ROM, das wir bei unserer anfänglichen Speicherbegehung als 2. Stock im Bereich 53248 bis 57343 kennengelernt haben, ist davon meilenweit entfernt! 4096 liegt außerdem mitten in einem Bereich, der ständig von Basic-Programmen überschrieben wird.

Die genaue Lösung des Rätsels soll erst in einer der nächsten Folgen gegeben werden. Aufgrund einer technischen Eigenart des VIC-II-Chips werden vom Zeichen-ROM zwei Geisterbilder im Bereich 4096 bis 8191 und im Bereich 36864 bis 40959 erzeugt. Der VIC-II-Chip »meint«, er hole seine Zeichen-Muster aus diesen Bereichen. In Wirklichkeit bezieht er sie im Normalfall immer aus dem Zeichen-ROM. Das ist eine Eigenart, die so recht in Alices Wunderland paßt!
paßt!

Wie sehen diese Zeichen-Muster aus? Auch dazu können Sie das Programm »SpeiLu« benutzen. Wenn Sie sich damit beispielsweise mal das A ansehen, finden Sie ein Muster, wie es in Bild 8 dargestellt ist.

Dieses 8 x 8-Gitter (auch Matrix genannt) enthält also das Abbild des Zeichens A. Alle Zeichen sind auf diese Weise als Punktmuster gespeichert in jeweils acht Speicherzellen (hier also von 53256 bis 53263). Ein dunkles Feld bedeutet ein gesetztes Bit (=1; im Zimmer ist etwas drin), ein helles Feld ein gelöschtes Bit (=0; das Zimmer enthält nichts).

Das Zeichen-ROM hat an nullter Stelle von 53248 bis 53255 das Zeichen mit dem Bildschirmcode 0 (den Klammeraffen **@**), an erster Stelle von 53256 bis 53263 — wie wir sehen — das Zeichen mit dem Bildschirmcode 1 (also das A) und so weiter nacheinander in Form von je acht Bytes als Bit-Muster gespeichert. Wenn Sie im Handbuch die Seite 133 f. aufschlagen, dann können Sie die Tabelle 2 mit dem Inhalt des Charakter-ROMs besser verstehen.

Probieren Sie mal aus, sich die einzelnen Zeichen mit dem Programm »SpeiLu« durch Eingabe der Bildschirmcodes (im Handbuch bis 127 als Pokes bezeichnet) abbilden zu lassen.

Das Programm »SpeiLu« (der Name kommt von »Speicher-Lupe«) enthält noch einige für Sie bislang noch geheimnisvolle Einzelheiten: die Hexadezimal- und die Binärzahlen, das Interrupt-System, das Kopieren des Zeichen-ROMs. Dies alles hängt zusammen mit der Frage: Wie kann man sich eigene Zeichen bauen und verwenden? Wir werden sie in der nächsten Folge gemeinsam beantworten.

Mit dem bisher zurückgelegten Weg durch das Dornengestrüpp sind wir unserem Ziel, der hochauflösenden Grafik, schon ein ganzes Stück nähergekommen. Ich hoffe, daß Sie nach der Ruhepause bis zur nächsten Folge die zweite Etappe der Expedition zu Dornröschen zusammen mit mir durchführen werden.

(Heimo Ponnath)

Wir suchen

die Anwendung des Monats

Anwendung des Monats, was ist das? Nun, Sie haben einen Commodore 64 oder einen VC 20 und versuchen diesen irgendwie sinnvoll einzusetzen. Unter einer sinnvollen Anwendung versteht die 64'er Redaktion alles, was beispielsweise Programme im häuslichen Bereich bewirken. Es kann sich dabei um die Berechnung der Benzinkosten für Ihren Wagen handeln, um ein eigenes Textverarbeitungsprogramm gehen, sich um die Verwaltung Ihrer Tiefkühltruhe drehen oder ein ausgeklügeltes Telefon- und Adreßregister sein.

Setzen Sie Ihren VC 20/C 64 mehr oder weniger beruflich ein? Auch, oder vor allem, das ist eine sinnvolle Anwendung. Sie führen die Lohn- und Gehaltsabrechnung, Ihre Lagerverwaltung, die Bestellungen auf einem Commodore-Heimcomputer durch? So spezielle Anwendungen wie die Berechnung der Statik von selbstgezimmerten Regalen, von Klimadiagrammen oder Vokabellernprogrammen für den Schulunterricht oder die Zinsberechnung bei Krediten sind ebenfalls Themen, die mehr als konkurrenzfähig sind.

Uns ist die Anwendung des Monats

500 Mark

wert

Schreiben Sie uns was Sie mit Ihrem Computer machen:

Redaktion 64'er
Aktion: Anwendung des Monats
Hans-Pinsel-Straße 2
8013 Haar bei München

Die notwendigen Informationen, wie Sie Ihr Programm einsenden müssen, sind dem Beitrag »Wie schicke ich meine Programme ein?« zu entnehmen.

Sprites

Ein Wort, ein Begriff, der zum ersten Mal durch das Erscheinen des Commodore 64 auf dem Markt geprägt wurde. Sprites oder MOBs (Moveable Object Blocks) sind charakteristisch für den C 64. Mittlerweile gibt es Heimcomputer, die mehr Sprites (bis zu 32) auf dem Bildschirm umherschwirren lassen. Dennoch, der Commodore 64 hat Pionierarbeit geleistet. Nicht nur deshalb suchen wir

das schönste Sprite

Sprites bestehen aus einer 21 x 24-Matrix, das sind insgesamt 504 Punkte. Also 504

1000 Mark **für das schönst**

Schicken Sie Ihr schönstes Sprite an:

Redaktion 64'er, Wettbewerb: Das schönste Sprite, Hans-Pinsel-Straße

Einmal im Monat gibt es die Superchance

Diese nicht einmalige Gelegenheit sollten Sie nutzen. Wie? Schicken Sie uns Ihr bestes, selbst erstelltes Programm. Bei der Art des Programms sind wir nicht wählerisch.

Sie haben ein sehr gutes (Schieß-, Knobel-, Denk-, Action-, Abenteuer-) Spiel geschrieben: einschicken!

Sie verfügen über ein komfortables Disketten-Kopier-(Sortier-)Programm mit einigen außergewöhnlichen Leistungsmerkmalen: einschicken!

Sie haben das Basic um einige sinnvolle Befehle erweitert: einschicken!

Sie arbeiten mit einem selbsterstellten Textverarbeitungsprogramm, einer eigenen Tabellenkalkulation, einem semiprofessionellen Datenverwaltungsprogramm: einschicken!

Sie zeichnen und konstruieren mit einem selbsterstellten Programm in hochauflösender Grafik: einschicken!

Wir freuen uns über jeden Beitrag und honorieren mit bis zu

2000 Mark für das Listing des Monats

Aus den besten Listings, die veröffentlicht werden, sucht die 64'er Redaktion einmal im Monat das »Listing des Monats« aus. Alle Listings, die im 64'er abgedruckt sind, werden mit 100 bis 300 Mark honoriert. Die genaue Vorgehensweise beim Einsenden von Listings ist in »Wie schicke ich meine Programme ein?« beschrieben.

Schicken Sie Ihr Listing an:

Redaktion 64'er
Superchance: Listing des Monats
Hans-Pinsel-Straße 2
8013 Haar bei München

verschiedene Punkte, die für die Gestaltung eines Sprites herangezogen werden können. Dazu kommen im Multicolor Modus noch mal drei verschiedene Farben, die einsetzbar sind.
Lassen Sie Ihrer Kreativität freien Lauf, es winken

e Sprite

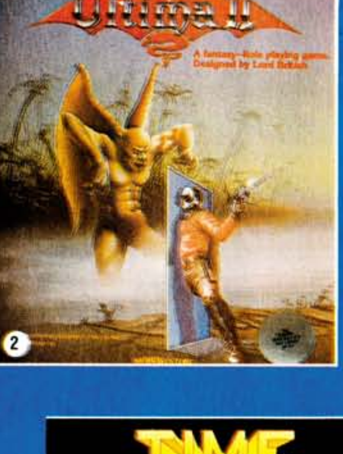
2, 8013 Haar bei München

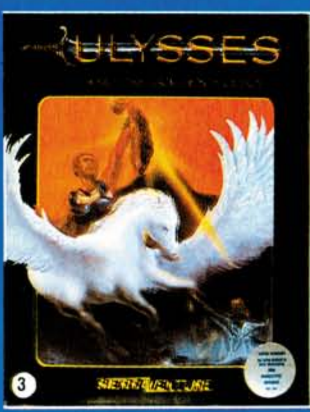


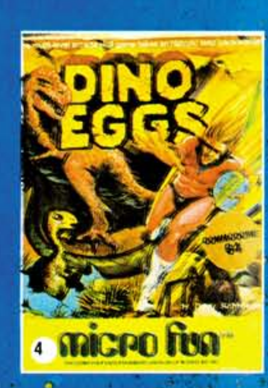




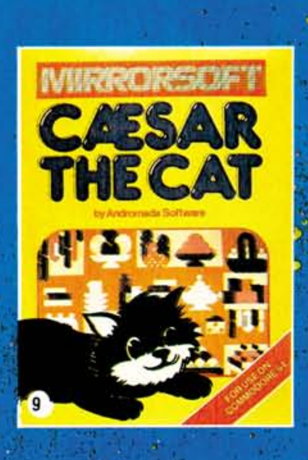


















Bitte beachten!

1: CRISIS MOUNTAIN Ein Vulkan bricht aus, als sich Terroristen in den Höhlen des Berges verstecken wollen. Sie verbergen gefährliche Bomben, die die ganze Welt in Schutt und Asche legen können. Sie sollen die Bomben suchen und entschärfen.

Atari 800, 48K **D MS111B DM 129,—**

2: ULTIMA II Abenteuer im Land der Fantasie — Ultima II ist eine Welt für sich. Dort gibt es weder Zeit- noch Raum-Grenzen. Piraten auf offener See, durch fremde Galaxien rasen, es gibt nichts, was Sie nicht erleben können.

Atari 400/800 (48K) D MS103B, DM 198,—
Apple II, II+, IIe, 48K,
DOS 3.2 u. 3.3 **D MS103C, DM 198,—**

3: ULYSSES Viele mutige Männer versuchten das goldene Vlies in ihren Besitz zu bringen, keinem gelang es. Sie sollen das goldene Vlies finden und es ihrem König bringen. Ihr Reisegefährte ist das geflügelte Pferd Pegasus.

Atari 400/800 (48K) D MS104B, DM 129,—
Apple II, II+, IIe, 48K,
DOS 3.3 **D MS104C, DM 129,—**
IBM-PC mit 48K, Color
Graphic Adapter **D MS104E, DM 129,—**

4: DINO EGGS Sie wollten nur ins Mesozoikum zurück, um das prähistorische Leben zu studieren. Unglücklicherweise brachten Sie aus dem 21. Jahrhundert die Masern mit: Alle Dinosaurier wurden angesteckt. Wird so das Schicksal der Saurier besiegelt? Sie können es verhindern, indem Sie soviele Sauriereier als möglich rechtzeitig in Sicherheit bringen. Urschlangen und Riesenspinnen machen Ihnen dabei zu schaffen.

Commodore 64 **D MS113A, DM 129,—**
Apple II, IIe, 48K,
Joystick **D MS 113C, DM 139,—**

5: SUPER PIPELINE Sie sind der Vorarbeiter eines Rohrverlegetrupps. Sie müssen dafür sorgen, daß die Pipeline benutzbar bleibt. Ein teuflischer Saboteur versucht, Ihre Arbeit zu boykottieren. Er versucht, die Leitung zu zerstören und Sie daran zu hindern, große Ölmengen zu fördern. Mit deutscher Bedienungsanleitung.
Commodore 64 **K MS108A, DM 39,—**

Telefonischer Bestell-Service in der Bundesrepublik:





HAPPY SOFTWARE

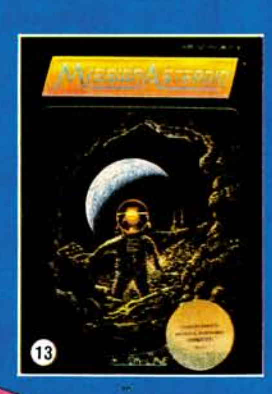
Jetzt gibt es Top-Programme und Spiele für Ihren Personal- und Homecomputer: Happy-Software bringt für die Leser von Happy-Computer die interessantesten Programme direkt von den Herstellern in England und USA auf den deutschen Markt.



11



12


13



14



15

Wir wollen Eure Spiele auf den Markt bringen! Wenn Sie ein Spiel schon programmiert oder eine Idee dazu haben, so lassen Sie uns das wissen. Zur ersten Kontaktaufnahme bitte Kurzbeschreibung oder Muster des Spieles an die Verlagsadresse von »Happy Software« senden. Wir reagieren sofort, zahlen hohe Lizenzgebühren und Bares im voraus.

Bestellkennzeichen: D = Diskette, K = Kassette

6: TIME ZONE Ramadu, der mächtige und skrupellose Herrscher des 1000 Lichtjahre entfernten Planeten Neburon, erklärt der Erde den Krieg. Der überlebende Teil der Menschheit würde vom teuflischen Ramadu versklavt werden.
Szenenwechsel: Sie sehen ein eigenartiges Fluggerät. Sie gehen darauf zu und schauen hinein, gehen durch eine große Einstiegsluke und nehmen Platz. Alles um Sie herum zu verschwindet, und das größte Abenteuer Ihres Lebens beginnt.

Apple II, II+, IIe, 48K, DOS 3.2 u. 3.3 — **D MS102C, DM 278,—**

7: JAMMIN Durchstreifen Sie 20 Irrgärten, sammeln Sie dabei alle Instrumente auf, die Sie finden können, und bringen Sie sie nach Hause. Wichtig ist, die gesammelten Geräte unterwegs nicht wieder zu verlieren oder sie zu zerstören. Mit deutscher Bedienungsanleitung.

Commodore 64 — **K MS109A, DM 39,—**

Multiplan (deutsch)
Ohne Abbildung:
Commodore 64 — **D MS115A, DM 336,—**

8: HOMEWORD Ein leistungsfähiges Textverarbeitungssystem. Sie können Ihre Korrespondenz umfassend gestalten, Texte in beliebiger Art und Weise formatieren. Seiteneinteilung frei wählbar. Randausgleich, Einrücken, Suchen und Ersetzen von Wörtern und vieles mehr kann HOMEWORD. Audio-Kassette und Handbuch erleichtern den Einstieg.

Commodore 64 mit Audiokassette — **D MS105A, DM 239,—**

9: CAESAR THE CAT Caesar ist eine Katze, die die Aufgabe hat, einen Lebensmittelvorrat vor Mäusen zu schützen. Sie müssen Caesar den Weg durch das Lager zeigen und ihm sagen, wann er nach einer Maus springen muß.

Commodore 64 — **K MS107A, DM 49,—**

10: MINER 2049er Gehen Sie auf die wildeste und verwegenste Jagd, die es je gab. Bevor Sie den berüchtigten Yukon Yokan fangen können, müssen Sie ihn durch zehn Stockwerke voller Fallen und tödlicher Herausforderungen jagen.

Apple II, IIe, Paddles ohne Joystick — **D MS 112C, DM 139,—**
IBM-PC, 64K, Spiele-, Farbadapter, Farbmonitor — **D MS112E, DM 149,—**

11: QUICK THINKING Zwei Spiele mit vielen Effekten und Musik: Bei ROBOT TABLES bedienen Sie eine Maschine, die Roboter baut. Abhängig vom Ergebnis einer Rechenaufgabe, erzeugt die Maschine Roboter oder Schrott.
Mit SUM VADERS eliminieren Sie nur mit Ihrer Intelligenz und der Geschicklichkeit Ihrer Finger aus dem All eindringende Roboter. Werden Sie die Roboterinvasion überleben?

Commodore 64 — **K MS106A, DM 39,—**

12: WIZARD AND THE PRINCESS Retten Sie die schöne Prinzessin vor dem bösen Zauberer Harlin rettet. Er hat sie auf sein Schloß hinter den großen Bergen entführt. Sie müssen nun die magischen Kräfte des Zauberers überwinden.

Commodore 64 — **D MS100A, DM 98,—**
Atari 400/800 (40K) — **D MS100B, DM 98,—**

13: MISSION ASTEROID Ein Asteroid rast auf die Erde zu. In wenigen Stunden wird er mit der Erde zusammenstoßen und eine Katastrophe auslösen. Sie sind der Astronaut, der die Erde retten kann.

Commodore 64 — **D MS101A, DM 129,—**
Atari 400/800 (40K) — **D MS101B, DM 129,—**
Apple II, II+, IIe, 48K, DOS 3.2 u. 3.3 — **D MS101C, DM 139,—**

14: PROFESSIONAL BLACK JACK Das einzige Spiel, bei dem der Spieler eine reale Gewinnchance hat. Alle Spielzüge werden von interessanten Musikeffekten begleitet. Holen Sie sich die Welt des Casinos nach Hause.

Atari 400/800/1200/Commodore 64 1 Disk. — **D MS114A, DM 189,—**

15: COMMODORE 64 MASTERCODE ASSEMBLER Mehr als nur ein Assembler: Er unterstützt die Erstellung von Maschinenprogrammen für den COMMODORE 64. Mastercode besteht aus mehreren Teilen, die mit Hilfe der Menütechnik beherrscht werden. Extrem niedrige Eingewöhnung!

Commodore 64 — **K MS110A, DM 62,—**

Bitte verwenden Sie für Ihre Bestellung die Software-Bestellkarte am Ende des Heftes.
Bestellungen in der Schweiz bitte an M & T Vertriebs-AG, Alpenstrasse 14, 6300 Zug, (042) 22 31 55

Erklärung der Steuerzeichen

Da sich immer wieder Schwierigkeiten bei der Identifizierung der Steuerzeichen in Listings ergeben, sind hier alle Steuerzeichen für die Commodore-Drucker VC 1515, VC 1526, MPS 801 (dem Nachfolgemodell des VC 1525) und für den Printer/Plotter 1520 angegeben.

Funktion	VC 1520	
CLR	= s	S
HOME	= S	s
REVERS ON	= R	r
REVERS OF	= f	F
INST	= t	T
STOP	= C	c
RECHTS	=]	△
LINKS	= △	]
UNTEN	= Q	q
OBEN	= q	Q
SCHWARZ	= p	P
WEISS	= E	e
ROT	= £	—
TUERKIS	= □	←
VIOLETT	= —	£
GRUEN	= ↑	π
BLAU	= ←	□
GELB	= π	↑
ORANGE	= a	A
BRAUN	= u	U
HELLROT	= v	V
GRAU 1	= w	W
GRAU 2	= x	X
HELLGRUEN	= y	Y
HELLBLAU	= z	Z
GRAU 3	= l	[
F1	= e	E
F2	= i	[illegible]
F3	= f	[illegible]
F4	= j	[illegible]
F5	= g	[illegible]
F6	= k	[illegible]
F7	= h	[illegible]
F8	= l	[illegible]

Großbuchstaben/Grafikmodus

Groß-/Kleinbuchstaben

Die Steuerzeichen des VC 1520 sind in den Listings unterstrichen, da der Printer/Plotter keine reversen Zeichen darstellen kann.

Wie schicke ich meine Programme ein?

Die 64'er-Redaktion freut sich über jeden Beitrag unserer Leser. Die Erfahrungen bei unseren Schwesterzeitschriften haben aber gezeigt, daß viele Einsender nicht genau wissen, in welcher Form sie ihre Manuskripte einsenden sollen. Die unten aufgeführten Punkte stellen keine »Richtlinien« dar. Dennoch sollte sich jeder, der ein Programm oder einen Artikel einsenden will, an ein gewisses Schema halten. Dies erleichtert zum einen die Arbeit der Redaktion, zum anderen kommt es auch Ihnen selbst zugute, da wir vollständige Listings oder Artikel schneller veröffentlichen können. Folgende Kriterien sind also generell zu beachten.

1. Auf der ersten Seite des Anschreibens sollten der Name, die vollständige Anschrift mit Telefonnummer sowie das Einsendedatum stehen.

2. In der »Betreffzeile« tragen Sie die genaue Spezifikation des verwendeten Computers und falls erforderlich, die Basic-, ROM- oder DOS-Versionen sowie die Speicherkonfigurationen ein. Der Titel des Artikels sollte ebenfalls daraus ersichtlich sein (auch für eventuelle Nachträge).

3. Im darauffolgenden Text können Sie Wesentliches zu Ihrer Person, zur Entstehungsgeschichte des Programms/Artikels, der Absicht, der Vorteile gegenüber anderen Programmen oder Methoden, der Eigenschaften und so weiter erläutern.

4. Auf der nächsten Seite beginnt die eigentliche Programmbeschreibung. Diese sollte nach Möglichkeit mit der Schreibmaschine geschrieben werden oder als Computerausdruck vorliegen. Den Text bitte mit mindestens eineinhalb oder doppeltem Zeilenabstand verfassen. Am linken und rechten Rand mindestens drei Zentimeter Freiraum für Korrekturen und Bemerkungen lassen.

5. Diese und alle nachfolgenden Seiten sollten durchnumeriert sein und in der Kopfzeile jeweils den Titel des Programms und den Namen des Autors enthalten.

6. Der Überschrift des Artikels schließen sich zwei oder drei einleitende Sätze an, welche die wesentlichen Punkte des Textes zusammenfassen.

Der Text selbst sollte in etwa folgenden Aufbau aufweisen:
— Angaben auf welchem Computer das Programm lauffähig ist sowie welche Erweiterungen und Peripherie notwendig sind
— ausführliche Beschreibung der Programmfunktion (mit Verweisen auf Ein-/Ausgabebeispielen wie Grafiken, Bildschirmfotografien, Hardcopys oder Diagrammen)
— detaillierte Programmbeschreibung (mit Verweisen auf Programmablaufplan, Variablendefinition, Startadressen der einzelnen Unterprogramme, Beschreibung wichtiger Programmzeilen etc.)
— eventuelle Umsetzung auf andere Basic-Dialekte oder Computer

7. Die genauen Lade- und Abspeicherschritte des Programms und der im Programm vorkommenden Routinen sollten dokumentiert sein.

8. Listings aus reprotechnischen Gründen nur als Original (keine Kopien) auf weißem, unliniertem Papier mit neuwertigem Farbband gedruckt einsenden. In den Listings dürfen grundsätzlich keine handschriftlichen Eintragungen stehen.

9. In den Kopfzeilen des Programms bitte den Titel desselben, die Computerkonfiguration, den eigenen Namen und die Adresse mit Telefonnummer eintragen (es soll vorkommen, daß sich Listings und Manuskripte verselbständigen, und mit beiden allein läßt sich wenig anfangen).

REM-Zeilen im Programm dienen der Übersichtlichkeit und sollten, falls nicht speicherkritische Aspekte dagegensprechen, immer zur Strukturierung eingesetzt werden (siehe u. a. »Sauberes Programmieren«).

10. Um das Eintippen für andere zu erleichtern, sollten Sie CHR$(X)-Werte und TAB(X) oder SPC(X) anstatt Cursor-Manipulationen für die Ausgabeformatierung verwenden. So ist die Befehlssequenz FOR I=1 TO 6:PRINT:NEXT zur Erzeugung von sechs Carriage Returns leichter einzutippen und auf andere Basic-Computer wesentlich einfacher zu übertragen. Und ist es nicht auch übersichtlicher statt einem Dutzend Cursor-Rechts-Symbolen einfach SPC(12) zu benutzen? Überprüfen Sie Ihr Programm einmal hinsichtlich dieser »Kleinigkeiten«.

11. Da wir (in Ihrem eigenen Interesse) nur getestete Programme veröffentlichen wollen, legen Sie bitte unbedingt eine Diskette oder Kassette, auf der das betreffende Programm mit mindestens einer Sicherheitskopie abgespeichert ist, bei. Auf der Diskette/Kassette und deren Umhüllung unbedingt den Namen mit vollständiger Adresse und Computerbezeichnung vermerken.

12. Wollen Sie mehrere Programme/Artikel gleichzeitig einsenden, so trennen Sie die Programme/Artikel nach dem oben aufgezeigten Schema. Die Einsendung mehrerer Disketten/Kassetten ist hingegen nicht notwendig.

13. Artikel können beliebig lang sein — von einzeiligen Routinen bis zu Serien über mehrere Ausgaben. Ein durchschnittlicher Artikel hat rund vier bis acht Schreibmaschinenseiten.

14. Hardcopys, Flußdiagramme, Zeichnungen und Bildschirmfotos dienen der Anschaulichkeit. Sie sollten nach Möglichkeit nicht fehlen. Zu jedem der vorgenannten »Zugaben« gehört aber eine Bildunterschrift und ein Verweis im Text.

15. Programme/Artikel die unserem Verlag zur Veröffentlichung angeboten werden, sollten aus urheberrechtlichen Gründen nicht gleichzeitig einem anderem Verlag vorliegen.

16. Das 64'er Magazin zahlt für Listings eine Pauschale zwischen 100 und 300 Mark. Für reine Artikel beträgt das Honorar zwischen 0,80 und 1,00 Mark pro Druckzeile. Für Disketten/Kassetten werden 30 Mark extra berechnet.

17. Sollten sich nach Erhalt eines positiven Anwortschreibens noch irgendwelche Änderungen oder Verbesserungen des Programms ergeben haben, teilen Sie uns das bitte umgehend mit. In diesem Falle benötigen wir ein vollständig neues Listing mit entsprechendem Datenträger.

(aa)

Funkende Computer

Wer kennt das nicht: Man möchte verschiedenen Hobbys — in der eigentlich zu knapp bemessenen Freizeit — am liebsten immer gleichzeitig nachgehen. Beschäftigt man sich mit einer Sache, bleibt keine Zeit mehr für anderes, was man auch gern tut. Nicht so bei Helmut Isenberg; er hat seine zwei Lieblingsbeschäftigungen — Amateurfunk und Programmieren — in nahezu idealer Weise miteinander verknüpft.



Zwei Hobbys in der »Bastelecke« vereint: Amateurfunk und Computer

Mit seinem Commodore 64 und dessen kleinem Bruder, dem VC 20, verschönt er sich die Freizeit in vielfältiger und interessanter Weise. Er nutzt sie für Funkfernschreiben (RTTY: Radio-Teletype) zur Auswertung von VHF- und UHF-Wettbewerben, zur Übertragung von Programmen über Funk und zur automatischen Fernbedienung anderer Computer über Funk.

Durch den Einsatz des Commodore ist die Abwicklung des Funkfernschreibens für den begeisterten Amateurfunker wesentlich komfortabler geworden. Doch bis er diese Erfahrung machen konnte, war der Weg steinig. Zunächst suchte er lange nach einem entsprechenden Programm. Es sollte beim Funkfernschreiben bequem zu handhaben sein: einfache Bedienerführung und maximaler Schutz vor Fehlbedienung. Außerdem sollte es auf einer Hardware laufen, die sich mit dem Geldbeutel-Inhalt des Amateurfunkers in Einklang bringen ließ. Doch mit dem, was sich ihm anbot, war er alles andere als zufrieden. So faßte er vor etwa zwei Jahren den Entschluß, seine Anforderungen selbst zu erfüllen. Beruflich war er zwar schon seit längerem in der Computerei beheimatet, doch seitdem hat der Spaß am Programmieren und Austüfteln von Erleichterungen noch um einiges zugenommen.

Als Hardware bevorzugt er den VC 20, zum einen aufgrund des günstigen Preises und zum anderen wegen der serienmäßig vorhandenen RS232-Schnittstelle (im Userport), die die nötigen Anschlüsse für einen Datenaustausch (und das ist Fernschreiben ja) bietet. Als besonders wichtigen Vorzug stellt Helmut Isenberg die Stabilität des Computers in Verbindung mit Funk heraus: Der Commodore stört aufgrund seiner metallischen Innenauskleidung in keiner Weise den Funkverkehr und umgekehrt. Mit anderen Computermodellen haben Kollegen in seinem Funkamateur-Ortsverband schon sehr schlechte Erfahrungen gesammelt.

Informationen per Amateurfunk zu übermitteln ist ein Hobby, das äußerste Präzision und Einhalten von vereinbarten Kürzeln bedeutet. Wenn im folgenden derartige Abkürzungen vorkommen, folgt eine kurze Erläuterung ihrer Bedeutung in Klammern.

Breites Anwendungsspektrum

Die Auswertung von Wettbewerben vereinfacht Routinearbeiten bei dem sonst so abwechslungsreichen Amateurfunk-Hobby. Bei Funkwettbewerben muß der Amateurfunker sein Logbuch mit allen

gefahrenen QSOs (Funkgespräche) beim Wettbewerbbearbeiter einreichen. Die Eintragungen müssen in einer vorgeschriebenen Norm vorliegen. Bis vor einiger Zeit war es hinderlich, immer alles von Hand entsprechend sauber und ordentlich einzutragen: Entfernung der Funkgespräche und Punktzahl wurden aus Listen abgelesen, doppelte Verbindungen mußten herausgefunden und entwertet werden. Mit dem Programm zur Erfassung und Bearbeitung der QSO-Werte lassen sich diese Auswertungen um ein Vielfaches schneller und vor allem auch ohne Fehler erledigen. Hinzu kommt der saubere und übersichtliche Ausdruck der QSL-Karten (Bestätigungskarten über Funkverbindungen) mit einem angeschlossenen Matrix-Drucker.

Das Programm zur Datenübertragung per Funk basiert auf der Funkfernschreib-Software, die im Verlaufe dieses Artikels noch eingehend beschrieben wird. Mit 300 Baud im ASCII-Format ist für Amateure auf dem 2 m UKW-Band ein optimiertes Verhältnis von Geschwindigkeit und Störabstand gegeben.

Datenaustausch ist auch das Stichwort bei der weiteren Anwendung: Rechnerkommunikation per Funk. In dem Ortsverband rund um Helmut Isenberg ist zwar der VC 20 am stärksten vertreten, doch auch mit Modellen anderer Hersteller wie Apple und Atari klappt die Zusammenarbeit einwandfrei.

Funkfernschreiben ist am interessantesten

Als eindeutig am interessantesten beurteilen die Amateurfunker das Funkfernschreiben. Das Hobby bekommt den Anstrich eines professionellen Datenaustauschs. Hierzu Helmut Isenberg: »Funkfernschreiben ist im Unterschied zu einfachem Amateurfunk kein 'Plapper-Funk'«. Nach seiner Schätzung arbeiten etwa 50 Prozent seiner Funkfernschreib-Kollegen noch mit alten ausgedienten Fernschreibern — ohne Computer und schwelgen dabei in einer Welle von Nostalgietechnologie. Mit einer Geschwindigkeit von 45 Baud werden die Fernschreiben über Funk an einen anderen Amateurfunker übermittelt. Die übliche Geschwindigkeit im deutschen Fernschreibnetz liegt bei 50 Baud. Doch die von den Funkern erworbenen Telex-Geräte sind größtenteils mechanisch zu abgenutzt und ausgeleiert, als daß sie bei 50 Baud noch mithalten könnten.

=RTTY Station DE DL4FBR DL4FBR DL4FBR=

Name	Helmut Helmut Helmut
QTH	Korbach Korbach Korbach
QTH	EL55E EL55E EL55E EL55E
DOK	F47 F47 F47 F47
RTTY	VC20, Softw. Homemade (RS232C)
Konverter	Filter (DJ6HP)
8FSK	CMOS-IC CD4016
Transceiver	IC-251E, QQE06/40, 120 Watt
Antenne	14 EL. Parabeam, 15M üb. Grund

Bild 1. Diese Werte müssen vor der ersten »richtigen« Nutzung des RTTY-Programms eingestellt werden.

Nicht allein in dem Funkamateur-Ortsverband, dem auch Helmut Isenberg angehört, stehen die Funkfernschreib-Programme RTTY-V3 und RTTY-C64 hoch in der Gunst der kommunikationsfreudigen Hobbyisten — bis hin nach Dänemark hat's bereits »gefunkt«. Die Arbeitsweise und die Funktionen beider Programme sind im Prinzip identisch. Für RTTY-V3 benötigt man einen VC 20 mit mindestens 8 KByte Speichererweiterung und für RTTY-C64 einen Commodore 64. Beide Programme sind in Basic geschrieben und für den Benutzer »offen«. Nach Ansicht von Helmut Isenberg kann sie ein Basic-kundiger Anwender leicht für eigene Belange modifizieren.

Hauptmenü

F1 — Senden
F3 — Empfang
F5 — Vorstellung
F7 — CQ-Ruf
F2 — Baudrate wählen
F4 — Test (RYRY)
F6 — Bandverarbeitung
F8 — Sonderfunktionen
E — Ende
Funktionstaste bitte

Bild 2. Alle Unterprogramme können über Funktionstasten aufgerufen werden.

Zuerst wird das Programm individualisiert

Vor der ersten »richtigen« Nutzung zum Senden oder Empfangen von Funkfernschreiben müssen einige benutzerspezifische Einstellungen vorgenommen werden. Dazu wird das Programm wie gewohnt geladen. Anzupassen sind das Rufzeichen, die Texte für die Stationsvorstellungen, die Gerätenummern für das Speichermedium (Kassette) etc. (Bild 1). Ist ein abgefragtes Gerät nicht vorhanden, so ist dies mit »0« zu kennzeichnen. Das so modifizierte Programm sollte unbedingt auf einer anderen Diskette oder einem anderen Band gesichert werden. So kann im Notfall immer auf die Originalversion zurückgegriffen werden.

Die gesamte Programmsteuerung erfolgt ausschließlich über Funktionstasten. Die Zusammenstellung der Funktionstastenbelegung ist auch gleichzeitig das Hauptmenü (Bild 2). Die von einem Untermenü ansteuerbaren Funktionen sind jeweils am Bildschirm angezeigt. Eine Besonderheit bildet die F8-Taste; über sie kann Helmut Isenberg jede angewählte Funktion und jedes Untermenü beenden.

Die Bedienung des Programms ist nach Helmut Isenbergs Erfahrungen »narrensicher« und auch für Ungeübte kein Problem. Nach dem Programmstart wird zunächst die Uhrzeit eingegeben. Anschließend ist die Übertragungsgeschwindigkeit festzulegen. Die Spanne reicht von 45 bis 300 Baud, wobei 45 Baud als Standard voreingestellt sind. Über die Funktionstaste F8 gelangt der Funker zum Hauptmenü (Bild 2). Hier kann er sich entscheiden, ob er Funkfernschreiben senden oder empfangen will.

Der Empfang funktioniert auch, wenn der Anwender selbst nicht anwesend ist. Dazu muß er den Computer eingeschaltet und das Programm geladen haben. Über eine PTT(Push-to-talk)-Leitung ist der Commodore mit dem Funkgerät verbunden. Will ein Kollege eine Nachricht übermitteln, so schaltet ein Relais dieser Leitung das

Programm auf Empfang und die Nachricht wird im Arbeitsspeicher hinterlegt. Die Information kann bis zu 8 KByte lang sein. Nachteil ist allerdings, daß eine hinterlegte Nachricht von einem nachfolgenden Sender überschrieben werden kann. Jeder, der ein Funkfernschreiben abschicken will und feststellt, daß der Empfänger gerade nicht persönlich anwesend ist, sollte also tunlichst vorher nachschauen, ob er nicht eine bereits vorhandene Information im Computer des anderen überschreibt.

Programmfunktionen

Und noch eines ist zu beachten: Bei der Umschaltung vom Sendebetrieb auf Empfang muß zuerst das Hauptmenü wieder aufgerufen werden. Die PTT bleibt dann so lange auf Sendung, bis alle Zeichen abgeschickt sind und der Sendende die F3-Taste betätigt. Auch wenn beispielsweise zehn Testschleifen gestartet werden, meldet sich zum Schluß das Hauptmenü. Es werden noch eine ganze Weile Fernschreibzeichen ausgegeben. Der Grund ist einfach: Alle Zeichen werden zunächst in einen 512 Byte großen Puffer geladen und das Unterprogramm ist beendet, bevor alle Zeichen mit einer Geschwindigkeit von 45 Baud gesendet sind. Während das Hauptmenü nach dem Untermenü »Senden« auf dem Bildschirm steht und noch Zeichen ausgegeben werden, kann logischerweise nicht auf »Empfang« geschaltet werden. Die Zeichen befinden sich in dem Puffer, in dem auch eingehende Informationen zwischengespeichert werden. Es ist jedoch möglich, in dieser Zeit jede Funktion aufzurufen, die für den Sendebetrieb des Funkgeräts nötig sind. So kann man unmittelbar nach den 10 CQ-Schleifen (CQ: Anruf an alle die gerade »hören«) die F1-Taste drücken und beispielsweise »PSE KKKK DE DL4FBR« (»Bitte komm für DL4FBR) eingeben. Danach betätigt man F8 (Sprung ins Hauptmenü), um dann mit F3 auf Empfang umzuschalten. Das Empfangsmenü wird jedoch erst nach dem letzten zu sendenden Zeichen aufgerufen und erst in diesem Moment wird auch die PTT-Leitung umgeschaltet.

Im folgenden werden die in Bild 2 zusammengestellten Programmfunktionen näher beschrieben.

F1 — Senden

Beim Sendebetrieb werden die eingegebenen Zeichen zunächst auf ihre Zulässigkeit geprüft, denn nicht alle Zeichen der Commodore-Tastatur lassen sich in dem für Funkfernschreiben notwendigen Baudot-Code darstellen. Die zugelassenen Zeichen werden über eine im Programm eingebundene Sendetabelle von ASCII nach Baudot umgewandelt. Dinge, die beim »normalen« Fernschreiber beachtet werden müssen, erledigt das Programm: Die Umschaltung von Buchstaben auf Ziffern erfolgt automatisch und nach jeweils 65 Zeichen wird ein Wagenrücklauf mit Zeilenvorschub gesendet. Damit die Empfangsseite sich sicher auf die übermittelten Zeichen einstellen kann, ist es möglich, am Anfang des Textes Buchstaben durch "=[" und Ziffern durch »=&« zu kennzeichnen.

Über die Funktionstaste F8 gelangt man zum Hauptmenü zurück, während über die PTT-Leitung weiter gesendet wird. Jetzt kann sofort die Vorstellung, CQ-Rufe (Anruf an alle) oder die Testschleife gewählt werden. Andere Funktionen — auch Empfang — sind erst möglich, wenn alle Zeichen gesendet sind.

F3 — Empfang

Die vom Funkgerät über den Konverter eingehenden Fernschreibzeichen werden im Computer mit Hilfe der Empfangstabelle in ASCII umgewandelt. Die Daten werden im Bereich 40960 bis 49152 (beim VC-20) beziehungsweise 32768 bis 40400 (beim Commodore 64) gespeichert und am Bildschirm ausgegeben. Seit kurzem ist es möglich, die Ausgabe aus dem Arbeitsspeicher auch direkt auf den Drucker zu leiten. Während des Empfangsbetriebs kann über die F3-Taste der Speicherpointer wieder zurückgesetzt werden. Der gespeicherte Text wird dann überschrieben. Das Textende wird in dem genannten Speicherbereich durch DEZ. 140 dargestellt. Bei Funkgesprächen mit mehreren Sende- und Empfangsperioden läßt sich immer wieder ein neuer Text anfügen, außer man drückt die F3-Taste während des Empfangsbetriebs.

Der Inhalt des Empfangsspeichers kann auf ein Speichermedium ausgegeben oder mit dem Unterprogramm »RTTY-UT« ausgedruckt werden. Ebenso wie beim Senden kommt man auch beim Empfang nur über das Hauptmenü in den jeweils anderen Modus.

F5 — Vorstellung

Hierbei kann man zwischen zwei Stationsvorstellungen auswählen: UKW und KW. Nach der Wahl schaltet die PTT-Leitung auf Senden und der Text wird übermittelt. Die Bildschirmausgabe geht zunächst sehr schnell bis der RS232-Puffer voll ist. Dann werden die Zeichen in Abhängigkeit von der Baudrate langsamer ausgegeben.

F7 — CQ-Ruf (Ruf an alle)

Nachdem eine Anzahl von Durchläufen desselben Inhalts festgelegt ist, wird über diese Funktion die PTT-Leitung auf Sendung geschaltet und der Text ausgegeben. Die Zahl der noch nachfolgenden Rufe wird mit gesendet und gibt der Empfangsstation die Information, wann der Sender auf Empfang schalten wird. Auch hier ist die Bildschirmausgabe — wie bei der Funktion »Vorstellung« am Anfang relativ schnell.

Am Ende wird bei der derzeitigen Programmversion nicht die übliche Abschlußbetätigung angefordert: »PSE KKK DE ... AT...«. Es ist angeraten, nach dem Durchlauf des letzten Textes direkt vom Hauptmenü auf »Senden« zu schalten und einen beliebigen Schlußtext zu übermitteln.

F2 — Baudrate wählen

Sieben verschiedene Baudraten sind vorgegeben und können über Funktionstasten angewählt werden. Neben diesen festen Werten läßt sich jede beliebige Geschwindigkeit einstellen, indem man auf der Adresse 666 verschiedene Werte mit dem Befehl »POKE« speichert.

Auch die Codierung ist vom Anwender veränderbar: Die ASCII-Baudot-Tabellen können gelöscht werden und das Programm ist so veränderbar, daß 8-Kanal ASCII empfangen/gesendet werden kann. Die gängigen Filterkonverter arbeiten jedoch nur bis 75 Baud einwandfrei. Die Erfahrung hat gezeigt, daß es möglich ist, über einfache Schaltungen gute Ergebnisse bis zu 600 Baud zu erzielen.

F4 — Test (RYRY)

Es ist in vielen Fällen ratsam, zunächst einen Testtext zu senden, bevor man die eigentliche Nachricht übermittelt. Häufig werden mehrere Testschleifen hintereinan-

der gesendet, um Sender und Empfänger optimal aufeinander einzustellen. Auch hier gilt: Die Bildschirmausgabe beim Empfänger ist so lange relativ schnell, bis der Puffer voll ist. Dann macht sich die Abhängigkeit der Ausgabe von der Baudrate bemerkbar. Am Ende dieses Untermenüs wird in das Hauptmenü verzweigt und die PTT sendet so weiter, bis alle Zeichen übermittelt und eine Funktionstaste gedrückt wurde, die den Empfangsmodus des angesteuerten Funkgeräts verlangt.

F6 — Bandverarbeitung

Mit diesem Untermenü ist es möglich, einen empfangenen Text auf Band zu speichern, einen beliebig langen Text auf Band vorzuschreiben und dieses Band direkt wieder in einem Funkfernschreiben auszugeben (Bild 3).

Die im folgenden angegebenen Funktionstasten beziehen sich auf dieses Untermenü.

Bandverarbeitung

F1 — QSO Sichern
F3 — Band Schreiben
F5 — Band Aussenden
F8 — Hauptmenü
Funktionstaste bitte

Bild 3. Die Bandverarbeitung ermöglicht das Senden und Empfangen von relativ umfangreichen Nachrichten.

F1 — QSO Sichern

Die während des Empfangs ankommenden und zwischengespeicherten Informationen können mit dieser Funktion auf Band gesichert werden. Weder durch das Kopien noch durch den Befehl »NEW« Wird der Text gelöscht und steht somit für andere Zwecke, wie Drucken, zur Verfügung. Beim Start des RTTY-Programms mit »RUN« wird die erste Stelle des Zwischenspeichers auf DEZ. 140 (normalerweise Kennzeichen für Textende) gesetzt. Somit entsteht bei Aufruf der Bandverarbeitung nach einem erneuten Programmstart der Eindruck, der Text sei gelöscht. Dieser Eindruck täuscht, denn er kann mit einer der Sonderfunktionen (F8) »Texte freigeben« wieder aktiviert werden. Die erste Speicherstelle wird hierbei allerdings auf »Blank« gesetzt und der jeweilige Textanfang ist verloren.

F3 — Band Schreiben

In diesem Programmteil kann man einen beliebigen Text am Bildschirm erfassen — beispielsweise für Rundschreiben. Die eingegebenen Zeichen werden direkt auf der Kassette gespeichert, der Text ist bei der gegenwärtig vorliegenden Programmversion leider noch nicht editierfähig.

F5 — Band Aussenden

Der auf dem Band gesicherte Text, sei er durch Textsicherung oder durch Vorschreiben entstanden, wird direkt ausgesandt. Die PTT-Leitung geht auf Sendung — wie immer so lange, bis alle Zeichen abgeschickt und eine Empfangsfunktion gemeldet wird. Während der Inhalt des Band-Puffers eingelesen wird (der Kassetten-Recorder-Motor läuft) wird die RS232-Ausgabe — Interrupt-gesteuert — angehalten. Die Ausgabe beginnt in dem Moment automatisch, wenn der Kassetten-Puffer voll ist und der Motor abschaltet.

F8 — Sonderfunktionen

Dieses Unterprogramm — aus dem Hauptmenü (Bild 2) anwählbar — stellt einen gewissen »Luxus« dar. Hier sind Fernschreibfunktionen vereint, die man im normalen QSO-Betrieb nicht ständig benötigt werden. Die wichtigste ist der SEL-CALL Betrieb, der es anrufenden Stationen, die das richtige Codewort kennen, ermöglicht, einen bis zu 8000 Zeichen langen Text (etwa 4 DIN A4 Seiten) abzuspeichern. Die übrigen Sonderfunktionen werden im folgenden näher erläutert.

Texte verwalten

Bei Aufruf dieser Funktion erscheinen vorgegebene standardisierte SEL-CALL-Textbausteine als »Platzhalter«. Sie können für einen Programmablauf geändert werden. Aus den mit Textnummern versehenen Bausteinen werden während des SEL-CALL-Betriebs die benötigten Texte automatisch zusammengestellt.

SEL-CALL Starten

Das Programm ist auf Empfang umgestellt und wartet, daß ankommende Fernschreibzeichen einem bestimmten Code entsprechen. Zum Beispiel: — EMPFANG VON XXX DE DC9QR DL4FBR — darauf meldet sich das System mit: HIER IST DL4FBR IM AUTOSTART GEBEN SIE IHRE MAX. 8KBYTE LANGE NACHRICHT EIN UND SCHLIESSEN SIE MIT ENDE — Die PTT-Leitung schaltet dann wieder auf Empfang und die anrufende Station kann ihre Nachricht hinterlegen. Wenn zum Schluß ENDE registriert wurde, meldet sich das Empfangssystem mit — HIER IST DL4FBR IM AUTOSTART VIELEN DANK FÜR IHRE NACHRICHT SIE KÖNNEN DEN TEXT ABRUFEN MIT DL4FBR? — Damit kann die anrufende Station testen, ob alles gut angekommen ist.

Speicher Senden

Mit dieser Funktion kann ein eingegangener Text wieder ausgesendet oder einfach nur am Bildschirm angeschaut werden.

Kurztext Eingeben

Ein maximal 3 KByte (beim Commodore 64 maximal 4 KByte) langer Text läßt sich mit dieser Funktion eingeben. Die jeweils letzten 10 Stellen können während der Eingabe korrigiert werden.

Zum Aussenden dieses Textes steht eine weitere Sonderfunktion »Kurztext Senden« zur Verfügung. Soll der Inhalt des Empfangsspeichers weiterversandt werden, so wird dies über die Sonderfunktion »QSO Senden« ermöglicht.

Texte Freigeben

Diese Funktion wurde bereits erwähnt: Nach »RUN« oder Wechsel der Baudrate wird die erste Stelle des Sende- und Empfangsspeichers auf dezimal. 140 (Textende) gesetzt, um den Speicher zu initialisieren. Werden die dort eingegebenen oder empfangenen Texte noch benötigt, so können sie mit dieser Funktion wieder aktiviert werden.

Zeit Stellen

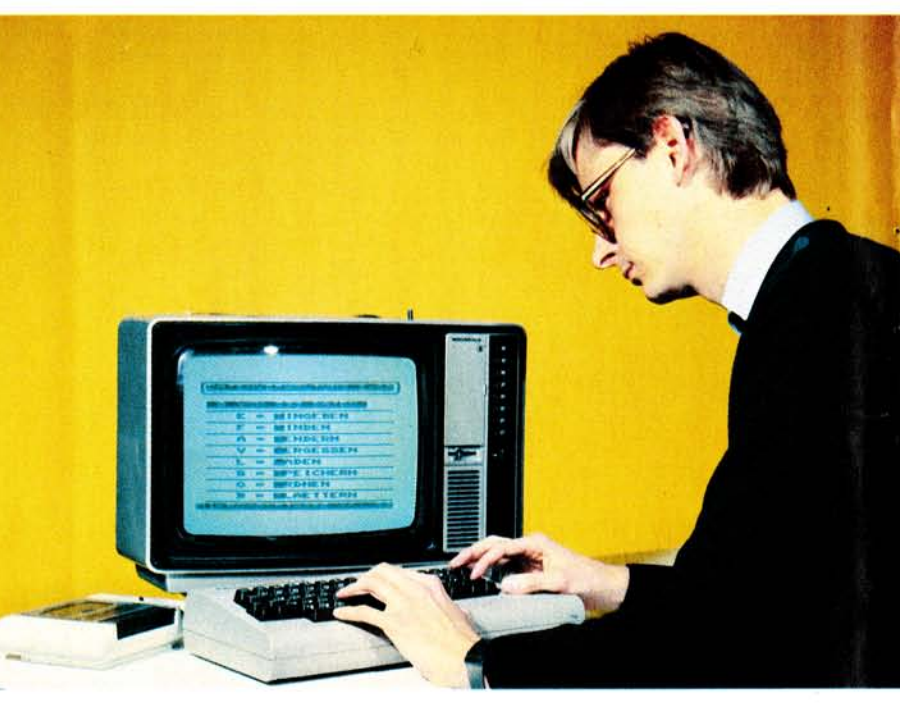
Diese Funktion wird in der Regel gleich nach dem Programmstart angesteuert. Wie bei »normalen« Fernschreiben ist es auch bei Funkfernschreiben üblich, die Nachrichten mit Tagesdatum und Uhrzeit zu versehen.

QRM (Störungen) Umschalten

Im Kurzwellenbetrieb können starke Störungen (beispielsweise schlechte Witterung) zu Empfangsproblemen bei der angesteuerten Station führen. Mit dieser Funktion ist es möglich, die allgemeine automatische Buchstaben/Ziffernumschaltung aufzuheben und vor jedem Zeichen die entsprechende Kennung anzugeben. Dadurch wird zwar die Sendezeit verdoppelt, aber die Lesbarkeit wesentlich erhöht.

(Helmut Isenberg/kg)

Klein aber oho — der VC 20



Das Hauptmenü: einfach und bedienungsfreundlich

```
0 IFPEEK(144)THENCLR:LOAD"DAT"
1 IFPEEK(650)<>128THENY=PEEK(56)-32:
  SYS(7821+256*Y)
2 POKE650,128:GOSUB1100
10 POKE36879,25:CLOSE1:U$="DATENVERWALTUNG 2.20"
   :GOSUB1000
13 PRINT" ■■"Z"■ VON"FR"■ DATEN ■":PRINT" "Y1$
20 PRINT"■■■■E = ■E■INGEBEN":PRINT" "Y1$
30 PRINT"■■■■F = ■F■INDEN":PRINT" "Y1$
40 PRINT"■■■■A = ■A■ENDERN":PRINT" "Y1$
50 PRINT"■■■■V = ■V■ERGESSEN":PRINT" "Y1$
60 PRINT"■■■■L = ■L■ADEN":PRINT" "Y1$
70 PRINT"■■■■S = ■S■PEICHERN":PRINT" "Y1$
74 PRINT"■■■■D = ■D■RDNEN":PRINT" "Y1$
75 PRINT"■■■■B = ■B■LAETTERN":PRINT" "Y1$
76 PRINT"■■MIKRO-SOFTWARE (C)'83■■"
79 P=PEEK(197):IFP=64GOTO79
80 IFP=8THENRUN
90 POKE198,0:IFP=49GOTO100:REME
91 IFP=42GOTO200:REMF
92 IFP=17GOTO300:REMA
93 IFP=27GOTO400:REMV
94 IFP=21GOTO500:REML
95 IFP=41GOTO600:REMS
96 IFP=35GOTO700:REMB
98 IFP=52GOTO2000:REMO
99 GOTO79
100 IFAZ>0GOTO110
101 U$="FORMAT ANLEGEN (0@)":GOSUB1000
102 CLR:OPEN5,0:PRINT"ANZAHL DER ZEILEN■":
    INPUT#5,AZ:PRINT:CLOSE5:GOSUB1100
103 IFAZ=0THENRUN
104 FORI=0TOAZ-1:PRINT"■■"I+1"■.SPALTENTITEL■":INPUT#5,X$:IFLEFT$(X$,1)="@"THEN
RUN
106 PRINT:T$(I)="■"+X$:NEXT
110 U$="DATEN EINGEBEN (@)":GOSUB1000
112 Z=Z+1:PRINT"SATZNUMMER :"Z:PRINT"NOCH"FR-Z"DATEN FREI"
114 FORI=0TOAZ-1:PRINT"■■"T$(I)"■■(@)"
115 INPUT#5,X$:PRINT
116 IFLEFT$(X$,1)="@"THENI=AZ:NEXTI:Z=Z-1:GOTO110
117 A$(Z,I)=X$:I$(I)=X$
119 NEXT
120 IFA$(Z,0)=""THENZ=Z-1
125 A(Z)=Z:SO=0
130 GOTO10
200 IFZ=0ORAZ=0THEN10
210 U$="DATEN FINDEN (@)":GOSUB1000
220 BG$="@":PRINT"SUCH-BEGRIFF■":PRINTBG$"■■":PRINT:INPUT#5,BG$:PRINT:IFBG$="@"G
OTO10
230 L=LEN(BG$):PRINT"■"
240 FORN=1TOZ:PRINT"■"N,BG$
245 FORM=0TOAZ-1
247 S=@INST(A$(N,M),BG$)
250 IFS=0ORS>LEN(A$(N,M))THEN258
251 PRINT"■■"N,BG$:FORI=0TOAZ-1:PRINT"■■"T$(I)"■"
252 PRINTA$(N,I):NEXT:GOSUB900
253 IFX$="N"ORX$="@"GOTO10
254 GOTO263
258 GETX$:IFX$="@"GOTO10
260 NEXTM
263 NEXTN:PRINT"■■■■■■■ ENDE DER DATEN "
265 GOSUB900
280 GOTO10
300 N=0:IFZ=0ORAZ=0GOTO10
310 U$="DATEN AENDERN (0@)":GOSUB1000
320 N=N+1:PRINTTAB(10)N"■":PRINT"SATZNUMMER ";:INPUT#5,X$:N=VAL(X$):PRINT:IFN>ZO
RN<1GOTO10
330 PRINT"■■■■■":FORI=0TOAZ-1:PRINT"■■"T$(I)
340 PRINT"■"A$(A(N),I):X=LEN(A$(A(N),I)):X=FNZ(X):PRINTLEFT$("■■■■■■■■",X)
341 INPUT#5,X$:PRINT
342 IFLEFT$(X$,1)="@"THENI=AZ:NEXTI:GOTO330
343 A$(A(N),I)=X$
344 NEXTI
345 GOSUB900:IFX$<>"N"GOTO310
350 GOTO10
400 IFZ<1GOTO10
405 U$="DATEN VERGESSEN (0)":GOSUB1000
420 PRINT"SATZNUMMER 0■";:INPUT#5,X$:PRINT:N=VAL(X$)
421 IFN>ZORN=0GOTO10
422 PRINT"■":GOSUB1000:FORI=0TOAZ-1:PRINT"■■"T$(I)"■":PRINTA$(A(N),I):NEXT
423 PRINT"■■■ VERGESSEN ?   (J/N) ■":GOSUB910:IFX$<>"J"GOTO400
430 FORI=0TOAZ-1:A$(A(N),I)=A$(Z,I):NEXT
433 FORI=1TOZ:IFA(I)<>ZTHENNEXT:STOP
436 A(I)=A(N):A=I:I=Z:NEXT
438 FORI=ATOZ:A(I)=A(I+1):NEXT
460 Z=Z-1:GOTO10
```

Basic-Listing von »Dat«

Man stelle sich vor: Jemand möchte am liebsten ein bequemes Auto fahren, weil er viel unterwegs ist, kann es sich aber momentan nicht leisten. Ein »Lebenskünstler« ist dann froh, wenn er überhaupt ein Auto besitzt. Ein Griesgram aber ärgert sich jeden Tag erneut, daß ihm nicht mehr vergönnt ist.

Walter Kröger ist in dieser Hinsicht eher ein Lebenskünstler. Bei ihm geht es nicht um ein Auto, sondern um einen Computer, den VC 20. Bei seinen täglichen Arbeiten in einer Münchner Baufirma ärgerte er sich lange darüber, daß Informationen über Angebote und Termine in Karteikästen nicht so abzulegen waren, daß er sie ohne Mühe unter jedem Stichwort zu jeder Zeit auffinden konnte. Die Lösung lag nahe: Mit einem Computer — so stellte er sich vor — geht das bestimmt besser.

```
500 U$="DATEN LADEN (@?)":GOSUB1000
502 CLR:GOSUB1120
505 PRINT"NAME██":PRINT"?█";:INPUT#5,N$:IFN$="@"GOTO10
507 IFN$="?"THENN$=""
510 PRINT"██":OPEN1,1,0,"FILE":INPUT#1,B$:PRINT"DATEI █"B$:FR=INT(FRE(0)/40)
512 IFLEFT$(B$,LEN(N$))<>N$THENCLOSE1:GOTO510
513 N$=B$:INPUT#1,AZ:DIMT$(AZ):FORI=0TOAZ-1:INPUT#1,T$(I):NEXT:DIMA(FR),A$(FR,AZ
),I$(AZ)
514 I=0:Z=0:GOTO525
516 I=I+1:IFI>AZ-1THENPRINT"██"N$;LEFT$(BL$,21-LEN(N$))"□█":PRINTTAB(15)Z"█ ":A(
Z)=Z:GOTO525
520 INPUT#1,A$(Z,I):PRINTA$(Z,I):GOTO516
525 Z=Z+1:I=0:INPUT#1,A$(Z,I):PRINTA$(Z,I):IFA$(Z,I)="@"THENCLOSE1:Z=Z-1:GOTO10
527 GOTO516
600 IFZ=0ORAZ=0GOTO10
602 U$="DATEN SPEICHERN (@)":GOSUB1000
605 PRINT"NAME██":N$="":PRINTB$"□":INPUT#5,N$:IFLEFT$(N$,1)="@"ORN$=""THENN$="":
GOTO10
610 PRINT"███":OPEN1,1,1,"FILE"
615 PRINT#1,N$:PRINT#1,AZ:FORI=0TOAZ-1:PRINT#1,T$(I):NEXT
620 FORN=1TOZ
630 FORI=0TOAZ-1:IFA$(A(N),I)=""THENA$(A(N),I)=" "
635 PRINT#1,A$(A(N),I):PRINTA$(A(N),I):NEXTI
636 PRINT"██"BL$"□":PRINT"█"N$"██"TAB(15)N"█ ":NEXTN
640 PRINT#1,"@":CLOSE1:GOTO10
700 N=0:IFZ=0ORAZ=0GOTO10
710 N=N+1:IFN>ZTHENN=1
720 U$="BLAETTERN (+-@)":GOSUB1000
730 IFN<1THENN=Z
735 PRINT"█"SPC(5)N"█ ("A(N)"█ )"
736 IFSOTHENPRINT"SORT: "MID$(T$(SO-1),2)
740 FORI=0TOAZ-1:PRINT"█"T$(I)"■":PRINTA$(A(N),I):NEXTI:GOSUB900:IFX$="N"ORX$="@
"GOTO10
741 IFX$="-"ANDNTHENN=N-1:GOTO720
742 GOTO710
900 PRINT"██████████████████████      ██ WEITER ██ ? █"
910 GETX$:IFX$=""GOTO910
920 RETURN
1000 Y1$="——————————————————————":BL$="                      "
1001 IFLEN(U$)<19THENU$=" "+U$+" "
1005 PRINT"██╭"Y1$"╮|██"LEFT$(BL$,20)"██|╰"Y1$"╯█"
1050 PRINT"████"SPC(11-LEN(U$)/2)U$"████"
1060 BL$="                      ":RETURN
1100 REM INIT
1110 FR=INT(FRE(0)/40):DIMT$(AZ),A(FR),A$(FR,AZ),I$(AZ)
1120 DEFFNZ(X)=INT((X)/22)+2:OPEN5,0
1125 IFFR=0THENFR=INT(FRE(0)/40)
1130 GOTO1060
2000 IFZ<2GOTO10
2005 U$="DATEN ORDNEN (0)":GOSUB1000:M=Z
2010 PRINT"ORDNUNG NACH WELCHEM":PRINT"DATENTEIL ?█"
2020 FORI=0TOAZ-1:PRINTI+1"= █"T$(I)"■":NEXT
2030 PRINT"█AUSWAHL 0█";:INPUT#5,X$:PRINT:PRINT:A%=VAL(X$)
2035 A%=A%-1:IFA%<0ORA%>AZTHEN10
2045 SO=A%+1
2050 PRINTM"█ WARTEN □":M=0:FORN=ZTO1STEP-1
2060 J=N
2062 IFA$(A(N),A%)<A$(A(J-1),A%)THENJ=J-1:GOTO2062
2065 IFJ<>NTHENGOSUB2100
2070 NEXTN:IFMGOTO2050
2080 GOTO10
2100 X=A(N):A(N)=A(J):A(J)=X:M=M+1:POKE36879,RND(1)*3+25:RETURN
READY.
```

Basic-Listing von »Dat« (Schluß)

In der Firma fand er wenig Resonanz, dort gab es andere Pläne und seine Probleme paßten gar nicht dazu. So blieb ihm keine andere Wahl: entweder ein kleiner Computer, den er von seinen privaten Ersparnissen bestreiten konnte — oder gar keiner.

Walter Kröger entschloß sich zum Computer und erstand einen VC 20, an den er eine Datasette und ein Fernsehgerät anschloß. Ein Drucker wäre zwar nötig, aber doch zunächst eine zu große Investition gewesen.

Nun brauchte er noch ein Programm, und zwar eines, das nicht mehr und nicht weniger leistet, als Informationen jeglicher Art nach seinen Wünschen zu verwalten und mit dem er sich schnell all das auf dem Bildschirm zeigen lassen kann, was er wissen will. Dieses Programm schrieb sein Sohn, der über die Entscheidung des Vaters recht begeistert war.

»Dat« ist ein einfaches Programm, sowohl im Aufbau als auch in der Handhabung. Die Bildschirmaufnahmen vermitteln einen Eindruck von der leichten Bedienbarkeit. Auf einen kurzen Nenner gebracht: ein Dateiverwaltungsprogramm,

Auch zum Lernen geeignet

bei dem Kröger selbst die Kategorien bestimmen und benennen kann. Er baut sich — je nach Aufgabenstellung — verschiedene Masken auf. So speichert er Daten von Angeboten, die er Kunden unterbreitet, behält die Übersicht über seinen Terminkalender, und in der Freizeit nutzt er dieses Programm um seinen englischen Wortschatz zu erweitern: Er hat sich bereits ein kleines englisches Wörterbuch an-

Gut vorbereitet auf die Groß-EDV

gelegt. Die Daten sind einfach einzugeben und zu verändern. Sortieren und Selektieren ist nach jedem der selbstdefinierten Felder möglich. Eine gewisse Erleichterung bei der täglichen Arbeit hat ihm der Computer schon verschafft, aber mittlerweile wird dieser Vorteil zusehends geringer. Der Massenspeicher »Kassette« reicht schon längst nicht mehr für die angefallene Datenmenge aus. Außerdem sind die Zugriffsgeschwindigkeiten

```
0 REM SAVE"INSTRING"
160 X=7821:Y=PEEK(56)-32:X=256*Y+X:PRINT"██████████████WARTEN..."
170 FORN=0TO370:READA:IFA<0THENA=Y-A
180 POKEX+N,A:NEXT
230 POKE631,13:POKE632,13:POKE198,2:PRINT"███LOAD"CHR$(34)"DAT":PRINT"██████RUN█
";:END
1000 DATA169,141,133,55,169,-30,133,56,169,76,133,124,169,162,133,125
1010 DATA169,-30,133,126,96,201,64,240,36,201,58,176,247,76,128,0
1020 DATA230,122,208,2,230,123,96,165,122,208,2,198,123,198,122,96
1030 DATA32,173,-30,208,3,32,180,-30,160,0,177,122,96,164,123,192
1040 DATA2,240,214,169,0,133,155,133,81,32,194,-30,201,178,240,3
1050 DATA76,8,207,32,141,205,165,14,72,165,71,133,90,165,72,133
1060 DATA91,32,173,-30,160,5,32,115,0,217,242,-31,208,226,136,208
1070 DATA245,32,189,-30,32,139,208,32,143,205,32,223,-31,160,0,177
1080 DATA71,133,83,200,177,71,141,122,-31,141,157,-31,200,177,71,141
1090 DATA123,-31,141,158,-31,32,253,206,32,139,208,32,143,205,32,223
1100 DATA-31,160,0,177,71,133,86,200,177,71,141,119,-31,141,152,-31
1110 DATA200,177,71,141,120,-31,141,153,-31,32,197,-30,201,41,240,39
1120 DATA201,44,240,3,76,8,207,32,173,-30,32,158,205,32,141,205
1130 DATA32,137,209,165,101,133,155,197,83,144,2,176,60,32,197,-30
1140 DATA201,41,240,3,76,8,207,166,155,173,29,18,221,9,18,240
1150 DATA9,232,228,83,208,243,162,0,240,29,134,82,169,0,133,92
1160 DATA230,82,230,92,164,92,196,86,240,12,185,29,18,164,82,217
1170 DATA9,18,208,221,240,234,232,134,81,165,90,133,71,165,91,133
1180 DATA72,104,16,14,160,0,169,0,145,71,200,165,81,145,71,76
1190 DATA212,-31,169,0,133,98,164,81,132,99,162,144,56,32,73,220
1200 DATA166,71,164,72,32,215,219,104,104,104,104,32,173,-30,76,174
1210 DATA199,96,224,0,208,251,162,8,189,247,-31,32,210,255,202,208
1220 DATA247,160,40,76,58,196,40,84,83,78,73,71,78,73,82,84
1230 DATA83,63,36
READY.
```

Ladeprogramm von »Dat«



Sortierkriterien des »Angebots-Karteikastens«

Programmierbeschreibung zu »DAT«:

Die Daten der zu verwaltenden Datensätze liegen in A$(X,Y) und die Sortierfolge wird zur Vermeidung störender »garbage-collects« durch die Zeiger A(Z) bestimmt. Die Texte für die Bezeichnung der einzelnen Datenfelder stehen in t$(..).

A$(Z,AZ-1)	Z Datensätze mit AZ-1 Feldern
AZ	Anzahl der Felder pro Datensatz
T$(AZ)	AZ Feldbezeichnung
A(Z)	Zeigerfeld für aufsteigende Sortierfolge
SO	Feldnummer, nach der momentan sortiert ist
FR	Anzahl der maximal möglichen Datensätze
U$	Text für die Überschrift eines neuen Bildschirms

Programmteile

0- 10	Ladefehlerkontrolle und Init von »INSTRING«
10- 76	Menü-Maske
79	Tastaturabfrage (ohne »garbage-collect«)
80- 99	Programmverzweigung
100	Dateiformat bereits angelegt? (dann weiter bei 110)
101-106	Dateiformat anlegen, Feldzahl und -texte festlegen
110-130	Datenfelder eingeben, Datenzähler Z erhöhen
200-280	Daten finden (innerhalb der Felder mit (__inst)·
300-350	Daten ändern
400-460	Daten vergessen und Sortierfolge korrigieren
500-527	Daten von Kassette laden
	Format der »FILE«-Dateien: Dateiname Anzahl der Felder pro Datensatz AZ Feldbezeichnungen der Felder T$(0-AZ) Datenfeld 1,1 Datenfeld 1,2 ... Datenfeld 2,1 ... —
600-640	Daten speichern in »FILE«
700-742	Daten blättern gemäß A(..) zyklisch vor / rück
900-920 SUB	Warteschleife mit »WEITER« bis eine Taste gedrückt
1000-60 SUB	Überschrift U$ in neuem Bildschirm ausgeben
1100-30 SUB	Initialisierung des Speichers nach CLR
2000-80	Sortieren der Daten
2100 SUB	Swap von zwei Zeigern A(.)

Programmbeschreibung und Variablendefinition

enorm hoch. Ganz zu schweigen davon, daß die Information schwarz auf weiß als Listenausdruck noch mehr zur Übersicht beitragen kann.

Doch auch hier bleibt Walter Kröger eher »Lebenskünstler«: Er sieht jetzt aus der Rückschau den eigentlichen Gewinn gar nicht mehr in erster Linie in der Zeitersparnis, die ihm seine selbstinitiierte Übergangslösung eingebracht hat. Er legt vielmehr Wert darauf, daß er sich — dank VC 20 — gut auf den zukünftigen Einzug einer größeren EDV-Anlage in seine Arbeitsumgebung vorbereitet hat — und fachkundig mithalten kann, wenn sich andere über neue Technologien und innovative Rationalisierungsmaßnahmen unterhalten.

(Walter Kröger/kg)

Fortsetzung von Seite 13

Dieser Aspekt bezieht sich aber auch auf eine ausreichende schriftliche Dokumentation des Programms. Bei der Änderungsfreundlichkeit sollte sich das Programm an neue Wünsche oder Ideen leicht anpassen lassen.

Was bietet nun »64 für Profis« in dieser Hinsicht? Doch einiges; so wird der genaue Hergang von der Programmidee über den Programmentwurf, das Flußdiagramm (wobei das Struktogramm oder Nassi-Shneidermann sehr ausführlich behandelt wird), den Vorbereitungsarbeiten zum Programmieren wie Maskenentwurf und Dateientwurfsblatt, Variablenliste und einigen Tips zum Umgang mit Variablen beschrieben. Gezeigt wird außerdem an einem gut dokumentierten Beispiel, wie man auch in Basic strukturiert programmieren kann. Viele weitere Anwendungsbeispiele wie Lagerverwaltung, eine einfache Textverarbeitung oder eine Literaturstellenverwaltung helfen das theoretisch Gelernte sofort in die Praxis umzusetzen. Dabei kommen jeweils die drei wesentlichen Bestandteile eines Programms — die Dateneingabe, die Datenverarbeitung und die Datenausgabe auf Floppy und Drucker — zum Tragen. Am Schluß wird mit der Verwendung von Programmierhilfen am Beispiel von Master 64 noch etwas Eigenwerbung betrieben. Man wird's Dank der vorhergehenden Kapitel verzeihen. Dieses Buch kann guten Gewissens auch dem Anfänger empfohlen werden. Sicherlich wird er nicht sofort alles verwenden können (dazu fehlt die Erfahrung), aber er wird sich von Anfang an einen guten Programmierstil zulegen. Das ist extrem wichtig, denn ein einmal verkorkster Stil läßt sich nur sehr schwer wieder korrigieren. Der Fortgeschrittene weiß, nachdem er das Buch durchgearbeitet hat, wie die Profis arbeiten, ob er es beherzigt, steht in seinem Ermessen. (aa)

Inserentenverzeichnis

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Michael M. Pauly (py)
Stellv. Chefredakteur: Michael Scharfenberger (sc)
Redakteure: aa = Albert Absmeier (130), rg = Christian Rogge (278), gk = Georg Klinge
Redaktionsassistenz: Dagmar Zednik (237)
Layout: Leo Eder (Ltg.), Willi Gründl, Walter Höß, Cornelia Weber
Fotografie: Janos Feitser, Titelfoto: Alex Kempkens

Auslandsrepräsentation:
Schweiz: Markt&Technik Vertriebs AG, Alpenstrasse 14, CH-6300 Zug, Tel. 042-223155/56, Telex: 862329 mut ch
USA: M&T Publishing, 2464 Embarcadero Way, Palo Alto, CA 94303; Tel. 415-2424-0600; Telex 752351

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck und zur Vervielfältigung der Programmlistings auf Datenträger. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Herstellung: Klaus Buck (180)

Anzeigenleitung: Peter Schrödel (156)
Anzeigenverkauf: Alfred Reeb (211)

Anzeigenverwaltung und Disposition: Sylvia Dietl (171)

Anzeigenformate: ¹⁄₁-Seite ist 266 Millimeter hoch und 185 Millimeter breit (3 Spalten à 58 mm oder 4 Spalten à 43 Millimeter). Vollformat 297x210 Millimeter. Beilagen und Beihefter siehe Anzeigenpreisliste.

Anzeigenpreise: Es gilt die Anzeigenpreisliste Nr. 1 vom 1. Oktober 1983.
Anzeigengrundpreise: ¹⁄₁ Seite sw: DM 7400,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1000,-. Vierfarbzuschlag DM 3000,-. Plazierung innerhalb der redaktionellen Beiträge: Mindestgröße ¹⁄₂-Seite
Anzeigen im Einkaufs-Magazin: Die ermäßigten Preise im Einkaufs-Magazin gelten nur innerhalb des geschlossenen Anzeigenteils, der ohne redaktionelle Beiträge ist. ¹⁄₁-Seite sw: DM 5400,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1000,-. Vierfarbzuschlag DM 3000,-. **Anzeigen in der Fundgrube: Private Kleinanzeigen** mit maximal 5 Zeilen Text DM 5,- je Anzeige.
Gewerbliche Kleinanzeigen: DM 10,- je Zeile Text.
Auf alle Anzeigenpreise wird die gesetzliche MwSt jeweils zugerechnet.

Vertriebsleitung, Werbung: Hans Hörl (114)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Plieninger Straße 100, 7000 Stuttgart 80 (Möhringen), Telefon (0711) 72004-0

Erscheinungsweise: »64'er« erscheint monatlich, Mitte des Vormonats.

Bezugsmöglichkeiten: Leser-Service: Telefon 089/4613-238. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen. Das Abonnement verlängert sich zu den dann jeweils gültigen Bedingungen um ein Jahr, wenn es nicht zwei Monate vor Ablauf schriftlich gekündigt wird.

Bezugspreise: Das Einzelheft kostet DM 6,-. Der Abonnementspreis beträgt im Inland DM 72,- pro Jahr für 12 Ausgaben. Darin enthalten sind die gesetzliche Mehrwertsteuer und die Zustellgebühren. Der Abonnementspreis erhöht sich um DM 18,- für die Zustellung im Ausland, für die Luftpostzustellung in Ländergruppe 1 (z.B. USA) um DM 38,-, in Ländergruppe 2 (z.B. Hongkong) um DM 58,-, in Ländergruppe 3 (z.B. Australien) um DM 68,-.

Druck: St. Otto-Verlag, Bamberg.

Verantwortlich: Für redaktionellen Teil: Michael M. Pauly.
Für Anzeigen: Peter Schrödel.

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/4613-0, Telex 5-22052

Mitteilung gem. Bayerischem Pressegesetz: Aktionäre, die mehr als 25% des Kapitals halten: Otmar Weber, Ingenieur, München; Carl-Franz von Quadt, Betriebswirt, München. Aufsichtsrat: Dr. Robert Dissmann (Vorsitzender), Karl-Heinz Fanselow, Hans-Jochen Wolf.

Telefon-Durchwahl im Verlag:

Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089-4613 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.

2000 Hamburg



2400 Lübeck

Ingenieurbüro
CHRISTINE MOEBIUS
Computer & Software
Würen 38, 2350 Neumünster
Tel. 04321/71623
Resselweg 8, 2400 Lübeck
Tel. 0451/594206

5060 Bergisch Gladbach

commodore
Vertragshändler
NORBERT STELLBERG
Buchholzstraße 1,
5060 Bergisch Gladbach
Telefon 02202/35053

5063 Overath

commodore
Vertragshändler
NORBERT STELLBERG
Blindenaaf 36
5063 Overath
Telefon 02206/6644

5200 Siegburg

commodore
Vertragshändler
NORBERT STELLBERG
Luisenstraße 26
5200 Siegburg
Telefon 02241/66854

6074 Rödermark

HYLAND
Computersysteme · BTX
Datentechnik · Video
Hardware · Software · Beratung · Verkauf · Service
commodore
Vertragshändler
Dieburger Straße 63
Ober-Roden Am Bahnhof
6074 Rödermark — 06074/96999

6800 Mannheim

Ihr
commodore -Partner
für Rhein/Neckar + Rhein-Pfalz
GAUCH+STURM
Computersysteme + Textsysteme
Casterfeldstr. 74-76
6800 Mannheim 24
Tel. (0621) 852045, Teletex 6211912

7000 Stuttgart



- MDT-Micro-Computer
- Bildschirm-Textsysteme
- Speicher-Schreibmaschinen
- Software-Rund-Um-Service

Fritz Computer GmbH
Schulze-Delitzsch-Str. 22, D-7000 Stuttgart 80,
Tel. (0711) 78000230/238/239

7700 Singen

Ihr autorisierter
commodore-Fachhändler
Beratung ● Verkauf ● Service
Schellhammer
7700 Singen, Freibühlstr. 21-23,
Tel. 07731/82020, Telex. 793716

7730 VS-Villingen

MAIER
Datensysteme GmbH
commodore
Vertragshändler
- Beratung · Verkauf
- Service · Software
- Interface-Entwicklung

Tel. 07721/70322
Gnädlingstraße 5
7730 Villingen-Weilersbach

8000 München

schulz computer
Beratung - Verkauf - Kundendienst
Commodore - IBM - Epson
Hewlett-Packard - EDV-Zubehör
Microcomputer Bauteile -
Literatur
Schulz computer
Schillerstr. 22, 8000 München 2
Telefon 089/597330

commodore
COMPUTER
Büromaschinen
Benno Knöbel
8000 München 2, Dachauer Str. 109
Telefon 089/5240421

Bitte Karte an der Perforation heraustrennen

★ ★ ★ ★ ★ ★ 64'er ★ ★ ★ ★ ★ ★

Bestellkarte für ein Geschenk-Abonnement

Schicken Sie »64'er, Magazin für Computerfans«

ab ______________ (Monat)
als Geschenk-Abonnement an

Name/Vorname

Straße

PLZ/Ort

Dauer des Geschenk-Abonnements:
☐ bis auf Widerruf (mindestens jedoch 12 Hefte)
☐ limitiert auf ein Jahr (12 Hefte)

Ich bezahle nur DM 72,– für 12 Hefte. Es entstehen mir keine weiteren Kosten, Lieferung frei Haus, Zustellgebühren und Mehrwertsteuer sind im günstigen Geschenk-Abo-Preis bereits enthalten.

Ich wünsche folgende Zahlungsweise:

☐ **Gegen Rechnung** 12 Hefte jährlich DM 72,–. Bitte keine Vorauszahlung leisten. Rechnung abwarten.

☐ **Bequem und bargeldlos durch Bankeinzug**
(nach Rechnungsstellung über 12 Hefte DM 72,–)

Bankleitzahl

Konto-Nr.

Geldinstitut

Adresse des Bestellers: (zugleich Rechnungsanschrift)

Name/Vorname

Straße

PLZ/Ort

Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann.

X

Datum Unterschrift

4-84

★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★

★ ★ ★ ★ ★ ★ 64'er ★ ★ ★ ★ ★ ★

Sofort-Bestellkarte für ein persönliches Abonnement

Ich beziehe »64'er, Magazin für Computerfans« bisher noch **nicht** regelmäßig per Post. Deshalb bestelle ich »64'er« ab ______________ (Monat) für die Dauer eines Jahres und weiter bis zur Abbestellung* mit allen Vorteilen eines persönlichen Abonnements:

- Ich bezahle (im Inland) für 12 Hefte nur DM 72,–. (Auslandspreise s. Impressum)
- Es entstehen mir keine weiteren Kosten. Lieferung erfolgt frei Haus, Porto und Zustellgebühren übernimmt der Verlag
- Die Lieferung erfolgt frei Haus.

Name

Straße/Nr.

Vorname

PLZ/Ort

*Das Abonnement verlängert sich nur dann zu den dann jeweils gültigen Bedingungen um ein Jahr, wenn es nicht 2 Monate vor Ablauf schriftlich gekündigt wird.

Ich bezahle mein Abonnement:

☐ **bequem und bargeldlos durch Bankeinzug** (12 Hefte jährlich DM 72,–)

von meinem Konto Nr.

Geldinstitut

Bankleitzahl (vom Scheck abschreiben)

☐ **Nach Erhalt der Rechnung**

Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann.

X

Datum/Unterschrift

4-84

Dieses Angebot gilt nur in der Bundesrepublik Deutschland einschließlich West-Berlin

★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★

64'er

BUCHLADEN-BESTELLKARTE

Liefern Sie mir zum Ladenpreis und gegen Rechnung:

Anzahl	Bestell-Nr.	Titel	Einzel-Preis inkl. MwSt

Zuzüglich DM 3,- Versandkostenanteil. Bitte beachten: Es werden nur Festbestellungen berücksichtigt. Eine Rückgabemöglichkeit besteht nicht, Ausnahme nur bei Beschädigung.
Genaue Lieferanschrift umseitig nicht vergessen!

Datum

Unterschrift

4-84



SOFTWARE-BESTELLKARTE

Liefern Sie mir zum Ladenpreis und gegen Rechnung folgende Programme auf Kassette:

Anzahl	Bestell-Nr.	Titel	Einzel-Preis inkl. MwSt

Zuzüglich DM 3,- Versandkostenanteil. Bitte beachten: Es werden nur Festbestellungen berücksichtigt. Alle Programme werden nur auf Kassette **nicht auf Diskette** geliefert. Eine Rückgabemöglichkeit besteht nicht. Ausnahme nur bei Beschädigung. Genaue Lieferanschrift umseitig nicht vergessen!

Datum

Unterschrift

4-84

Verlags-Garantie

Sie erhalten »64'er, Magazin für Computerfans« ab der von Ihnen gewünschten Ausgabe

★

Lieferung erfolgt frei Haus inkl. Mehrwertsteuer. Die Zustellgebühren sind im günstigen Abonnementspreis bereits enthalten.

★

Es entstehen Ihnen keine weiteren Kosten

★

Das Abonnement verlängert sich nur dann um ein Jahr zu den dann jeweils gültigen Bedingungen, wenn Sie es nicht bis 8 Wochen vor Ablauf schriftlich kündigen.

★

Hans Hörl · Vertriebsleiter

Bitte frei- machen

Postkarte

Antwort

64'er

Magazin für Computerfans

LESER-SERVICE

Markt & Technik
Verlag Aktiengesellschaft
Hans-Pinsel-Straße 2

8013 Haar bei München

Verlags-Garantie

Der von Ihnen Beschenkte erhält »64'er, Magazin für Computerfans« ab der von Ihnen gewünschten Ausgabe

★

Lieferung erfolgt frei Haus inkl. Mehrwertsteuer. Die Zustellgebühren sind im günstigen Abonnementspreis bereits enthalten.

★

Es entstehen Ihnen keine weiteren Kosten

★

Das Abonnement verlängert sich nur dann um ein Jahr zu den dann jeweils gültigen Bedingungen, wenn Sie es auf dieser Bestellkarte bis auf Widerruf anfordern.

★

Hans Hörl · Vertriebsleiter

Bitte frei- machen

Postkarte

Antwort

64'er

Magazin für Computerfans

LESER-SERVICE

Markt & Technik
Verlag Aktiengesellschaft
Hans-Pinsel-Straße 2

8013 Haar bei München

Lieferanschrift

Liefern Sie bitte meine Bestellung an folgende Adresse

Name des Bestellers

Anschrift

PLZ Ort

Telefon

Bitte frei- machen

Postkarte

Antwort



Magazin für Computerfans

BUCHLADEN

Markt & Technik
Verlag Aktiengesellschaft
Hans-Pinsel-Straße 2

8013 Haar bei München

Lieferanschrift

Liefern Sie bitte meine Bestellung an folgende Adresse

Name des Bestellers

Anschrift

PLZ Ort

Telefon

Bitte frei- machen

Postkarte

Antwort

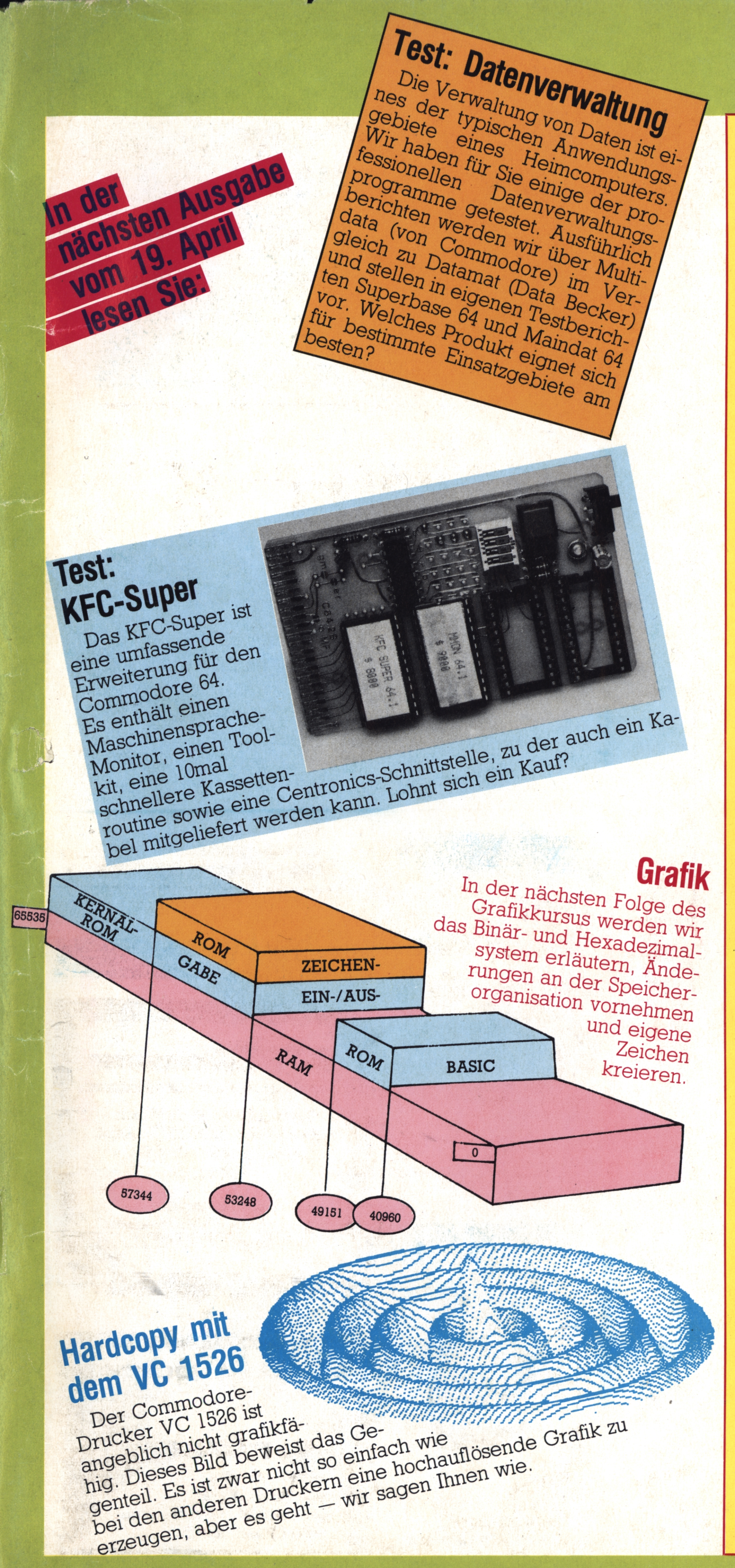
64'er

Magazin für Computerfans

BUCHLADEN

Markt & Technik
Verlag Aktiengesellschaft
Hans-Pinsel-Straße 2

8013 Haar bei München



VORSCHAU

Das DOS auf der Demo-Diskette

Auf der von Commodore zur Floppy 1541 mitgelieferten Demo-Diskette befindet sich ein kleines Programm, das sich DOS-5.1 nennt. Es erlaubt die bequeme Benutzung der verschiedenen Disketten-Kommandos. Doch welche Befehle sind das? Was bewirken Sie?

Wissenswertes über Schnittstellen

Sie arbeiten ständig mit ihm, dem seriellen Bus des VC 20 und Commodore 64. Wissen Sie genau wie er funktioniert? Wir haben uns bemüht, das Geschehen auf dem seriellen Bus so ausführlich wie möglich zu beschreiben.

Wir liefern auch Informationen, wie Sie Drucker mit anderen Schnittstellen wie Centronics oder V.24 an Ihrem Commodore 64 oder VC 20 betreiben können.

Strukturierte Programmierung

Was ist eigentlich ein Top-Down-Entwurf? Was ist der Unterschied zur Bottom-Up-Methode? Was ist ein Flußdiagramm oder ein Nassi-Shneidermann-Diagramm? Was ist ein Modul und worauf ist beim Erstellen eines Moduls zu achten? Wir zeigen es Ihnen an Beispielen.

Listings

— Spiele: Schatzsucher, Roulett, Fahrsimulator und Schmatzer
— kleines Adreß- und Telefonregisterprogramm
— Namen und die ID einer Diskette ändern, ohne die Programme zu löschen
— Mitglieder- und Beitragsverwaltungsprogramm
— dreidimensionale Grafik auf dem VC 20
— Batch-Copy: So sichern Sie Ihre Programme von der Diskette auf Band
— und natürlich wieder viele Tips und Tricks für den Commodore 64 und VC 20.

Mit dem RX-80 kommt der Commodore 64* erst richtig zum Ausdruck.

Der RX-80 ist ein preiswerter Matrixdrucker, der zum Commodore 64 gehört, wie der Halbleiterkristall zum Transistor. Weil er den richtigen Draht hat und 'ne Menge kann. Mit ihm bringen Commodore 64 Fans die Leistung ihres Computers makellos zu Papier. In 100 Zeichen pro Sekunde, auf 80 Zeichen pro Zeile. Daß der RX-80 128 Schriftarten und 10 internationale Zeichensätze parat hat und einfach zu bedienen ist, versteht sich fast von selbst. Auch, daß er den gesamten Zeichenvorrat des Commodore 64 verarbeitet. Schließlich kommt er von EPSON, dem erfahrensten Druckerhersteller der Welt. Das bürgt für Qualität und Zuverlässigkeit. Den RX-80 gibt es überall im Fachhandel.



EPSON

Technologie, die Zeichen setzt.

EPSON Deutschland GmbH · Am Seestern 24 · 4000 Düsseldorf 11 · Tel. (0211) 59520

Ausführliche Informationen mit Fachhändlernachweis, wenn Sie uns schreiben.

Name: ______________ Tel.: ______________

Firma: ______________

Straße: ______________

PLZ: ______ Ort: ______________

CF 1 RX

Interface RX-80/Commodore 64 ab sofort im EPSON-Fachhandel erhältlich. Preis: DM 129,–* (*unverbindliche Preisempfehlung inklusiv Mehrwertsteuer)

* Commodore 64 ist ein Warenzeichen der Commodore Business Machines Inc. USA.